Union internationale des télécommunications

UIT-T X.1141

(06/2006)

SECTEUR DE LA NORMALISATION
DES TÉLÉCOMMUNICATIONS
DE L'UIT

SÉRIE X: RÉSEAUX DE DONNÉES, COMMUNICATION ENTRE SYSTÈMES OUVERTS ET SÉCURITÉ

Sécurité des télécommunications

Langage de balisage d'assertion de sécurité (SAML 2.0)

Recommandation UIT-T X.1141

RECOMMANDATIONS UIT-T DE LA SÉRIE X

RÉSEAUX DE DONNÉES, COMMUNICATION ENTRE SYSTÈMES OUVERTS ET SÉCURITÉ

RÉSEAUX PUBLICS DE DONNÉES	
Services et fonctionnalités	X.1–X.19
Interfaces	X.20–X.49
Transmission, signalisation et commutation	X.50–X.89
Aspects réseau	X.90–X.149
Maintenance	X.150–X.179
Dispositions administratives	X.180–X.199
INTERCONNEXION DES SYSTÈMES OUVERTS	
Modèle et notation	X.200–X.209
Définitions des services	X.210–X.219
Spécifications des protocoles en mode connexion	X.220–X.229
Spécifications des protocoles en mode sans connexion	X.230–X.239
Formulaires PICS	X.240–X.259
Identification des protocoles	X.260–X.269
Protocoles de sécurité	X.270–X.279
Objets gérés des couches	X.280–X.289
Tests de conformité	X.290–X.299
INTERFONCTIONNEMENT DES RÉSEAUX	
Généralités	X.300–X.349
Systèmes de transmission de données par satellite	X.350–X.369
Réseaux à protocole Internet	X.370–X.379
SYSTÈMES DE MESSAGERIE	X.400–X.499
ANNUAIRE	X.500–X.599
RÉSEAUTAGE OSI ET ASPECTS SYSTÈMES	
Réseautage	X.600–X.629
Efficacité	X.630–X.639
Qualité de service	X.640–X.649
Dénomination, adressage et enregistrement	X.650–X.679
Notation de syntaxe abstraite numéro un (ASN.1)	X.680–X.699
GESTION OSI	
Cadre général et architecture de la gestion-systèmes	X.700–X.709
Service et protocole de communication de gestion	X.710–X.719
Structure de l'information de gestion	X.720–X.729
Fonctions de gestion et fonctions ODMA	X.730–X.799
SÉCURITÉ	X.800–X.849
APPLICATIONS OSI	
Engagement, concomitance et rétablissement	X.850–X.859
Traitement transactionnel	X.860–X.879
Opérations distantes	X.880–X.889
Applications génériques de l'ASN.1	X.890–X.899
TRAITEMENT RÉPARTI OUVERT	X.900–X.999
SÉCURITÉ DES TÉLÉCOMMUNICATIONS	**X.1000–**

Pour plus de détails, voir la Liste des Recommandations de l'UIT-T.

Recommandation UIT-T X.1141

Langage de balisage d'assertion de sécurité (SAML 2.0)

Résumé

SAML est un cadre de travail fondé sur XML pour l'échange d'informations de sécurité. Ces informations de sécurité sont exprimées sous la forme d'assertions sur des sujets, où un sujet est une entité (un humain ou un ordinateur) qui a une identité dans un certain domaine de sécurité. Une seule assertion peut contenir plusieurs déclarations internes différentes sur l'authentification, l'autorisation, et des attributs. La présente Recommandation définit un protocole par lequel les clients peuvent demander des assertions de la part des autorités SAML et obtenir d'elles une réponse. Ce protocole, qui consiste en formats de messages de demande et de réponse fondés sur XML, peut être lié à de nombreux protocoles sous-jacents de communication et de transport différents; SAML définit actuellement une liaison à SOAP sur HTTP. En créant leurs réponses, les autorités SAML peuvent utiliser diverses sources d'information, comme des mémoires et des assertions de politique externes reçues en entrée dans des demandes. La présente Recommandation définit les éléments, sujets, conditions, règles de traitement et déclarations d'assertions SAML. De plus, elle développe un profil de métadonnées SAML complet qui inclut un espace de nom associé, des types de données communs, des règles de traitement et un traitement de la signature. Plusieurs liaisons de protocoles comme SOAP, PAOS (SOAP inversé), HTTP redirect, HTTP POST, entre autres, sont aussi développées. La Recommandation fournit une liste complète des profils SAML tels que le profil de navigateur de la toile SSO et un profil unique de fermeture de session pour permettre une large adoption de SAML 2.0 dans l'industrie. Des lignes directrices pour le contexte d'authentification et la conformité sont également fournies.

La présente Recommandation est techniquement équivalente et compatible avec la norme OASIS SAML 2.0.

Source

La Recommandation UIT-T X.1141 a été approuvée le 13 juin 2006 par la Commission d'études 17 (2005-2008) de l'UIT-T selon la procédure définie dans la Recommandation UIT-T A.8.

AVANT-PROPOS

L'UIT (Union internationale des télécommunications) est une institution spécialisée des Nations Unies dans le domaine des télécommunications. L'UIT-T (Secteur de la normalisation des télécommunications) est un organe permanent de l'UIT. Il est chargé de l'étude des questions techniques, d'exploitation et de tarification, et émet à ce sujet des Recommandations en vue de la normalisation des télécommunications à l'échelle mondiale.

L'Assemblée mondiale de normalisation des télécommunications (AMNT), qui se réunit tous les quatre ans, détermine les thèmes d'étude à traiter par les Commissions d'études de l'UIT-T, lesquelles élaborent en retour des Recommandations sur ces thèmes.

L'approbation des Recommandations par les Membres de l'UIT-T s'effectue selon la procédure définie dans la Résolution 1 de l'AMNT.

Dans certains secteurs des technologies de l'information qui correspondent à la sphère de compétence de l'UIT-T, les normes nécessaires se préparent en collaboration avec l'ISO et la CEI.

NOTE

Dans la présente Recommandation, l'expression "Administration" est utilisée pour désigner de façon abrégée aussi bien une administration de télécommunications qu'une exploitation reconnue.

Le respect de cette Recommandation se fait à titre volontaire. Cependant, il se peut que la Recommandation contienne certaines dispositions obligatoires (pour assurer, par exemple, l'interopérabilité et l'applicabilité) et considère que la Recommandation est respectée lorsque toutes ces dispositions sont observées. Le futur d'obligation et les autres moyens d'expression de l'obligation comme le verbe "devoir" ainsi que leurs formes négatives servent à énoncer des prescriptions. L'utilisation de ces formes ne signifie pas qu'il est obligatoire de respecter la Recommandation.

DROITS DE PROPRIÉTÉ INTELLECTUELLE

L'UIT attire l'attention sur la possibilité que l'application ou la mise en œuvre de la présente Recommandation puisse donner lieu à l'utilisation d'un droit de propriété intellectuelle. L'UIT ne prend pas position en ce qui concerne l'existence, la validité ou l'applicabilité des droits de propriété intellectuelle, qu'ils soient revendiqués par un membre de l'UIT ou par une tierce partie étrangère à la procédure d'élaboration des Recommandations.

A la date d'approbation de la présente Recommandation, l'UIT n'avait pas été avisée de l'existence d'une propriété intellectuelle protégée par des brevets à acquérir pour mettre en œuvre la présente Recommandation. Toutefois, comme il ne s'agit peut-être pas des renseignements les plus récents, il est vivement recommandé aux développeurs de consulter la base de données des brevets du TSB sous http://www.itu.int/ITU-T/ipr/.

TABLE DES MATIÈRES

Recommandation UIT-T X.1141

Langage de balisage d'assertion de sécurité (SAML 2.0)

1 Domaine d'application

La présente Recommandation définit le langage de balisage d'assertion de sécurité (SAML, *security assertion markup language*) (SAML 2.0). SAML définit la syntaxe et la sémantique du traitement des assertions faites à propos d'un sujet par une entité système. Dans le cours de ces assertions, ou en s'appuyant sur de telles assertions, les entités systèmes SAML peuvent faire usage d'autres protocoles pour communiquer à propos de l'assertion elle-même, ou du sujet d'une assertion. La présente Recommandation définit la structure des assertions SAML, un ensemble associé de protocoles, et les règles de traitement impliquées par la gestion d'un système SAML.

Les assertions et messages de protocole SAML sont codés en XML et utilisent les espaces de nom XML. Elles sont normalement incorporées dans d'autres structures pour le transport, comme les demandes HTTP POST ou les messages SOAP codés en XML. La présente Recommandation spécifie aussi des liaisons SAML qui fournissent le cadre de travail pour l'incorporation et le transport des messages de protocole SAML. De plus, la présente Recommandation fournit aussi un ensemble de base de profils qu'utilisent les assertions et protocoles SAML pour traiter des cas d'utilisation spécifiques ou réaliser l'interopérabilité lors de l'utilisation de caractéristiques SAML.

La présente Recommandation définit ce qui suit:

1) exigences de conformité pour SAML;
2) assertions et protocoles pour SAML:
   - schéma d'assertions SAML;
   - schéma de protocoles SAML;
3) liaisons pour SAML;
4) profils pour SAML:
   - schéma de profil ECP de SAML;
   - schéma de profil d'attribut X.500/LDAP SAML;
   - schéma de profil d'attribut DCE PAC SAML;
   - schéma de profil d'attribut XACML SAML;
5) métadonnées pour SAML;
6) schéma de métadonnées pour SAML;
7) contexte d'authentification pour SAML.

2 Références normatives

Les Recommandations suivantes et autres références contiennent des dispositions qui par leur référence dans le présent texte, constituent des dispositions de la présente Recommandation. Au moment de la publication, les numéros d'édition indiqués étaient valides. Toute Recommandation et autre référence est sujette à révision, et les parties aux accords fondés sur la présente Recommandation sont invitées à rechercher s'il est possible d'appliquer l'édition la plus récente des Recommandations et autres références listées ci-dessous. Le Bureau de normalisation des Télécommunications de l'UIT tient à jour la liste des Recommandations de l'UIT-T. L'IETF tient à jour la liste des RFC, ainsi que de celles qui ont été rendues obsolètes par des RFC plus récentes. Le W3C, le Consortium Unicode et Liberty Alliance maintiennent à jour une liste des recommendations et autres publications les plus récentes.

– Recommandation UIT-T X.660 (2004) | ISO/CEI 9834-1:2005, *Technologies de l'information – Interconnexion des systèmes ouverts – Procédures opérationnelles des organismes d'enregistrement de l'OSI: procédures générales et arcs sommitaux de l'arborescence des identificateurs d'objet ASN.1.*

– Recommandation UIT-T X.667 (2004) | ISO/CEI 9834-8:2005, *Technologies de l'information – Interconnexion des systèmes ouverts – Procédures opérationnelles des organismes d'enregistrement de l'OSI: génération et enregistrement des identificateurs uniques universels (UUID) et utilisation de ces identificateurs comme composants d'identificateurs d'objet ASN.1.*

– Recommandation UIT-T X.680 (2002) | ISO/CEI 8824-1:2002, *Technologies de l'information – Notation de syntaxe abstraite numéro un: spécification de la notation de base.*

– Recommandation UIT-T X.800 (1991), *Architecture de sécurité pour l'interconnexion en systèmes ouverts d'applications du CCITT*.

– Recommandation UIT-T X.811 (1995) | ISO/CEI 10181-2:1996, *Technologies de l'information – Interconnexion des systèmes ouverts – Cadres de sécurité pour les systèmes ouverts: cadre d'authentification*.

– Recommandation UIT-T X.812 (1995) | ISO/CEI 10181-3:1996, *Technologies de l'information – Interconnexion des systèmes ouverts – Cadres de sécurité pour les systèmes ouverts: cadre de contrôle d'accès*.

– Recommandation UIT-T X.1142 (2006), *Langage de balisage extensible de contrôle d'accès (XACML 2.0)*.

– IETF RFC 1034 (1987), *Noms de domaine – Concepts et facilités*.

– IETF RFC 1510 (1993), *Demandeur d'authentification de réseau Kerberos(V5)*.

– IETF RFC 1750 (1994), *Randomness Recommendations for Security*. *(Recommandations de chiffrement aléatoire pour la sécurité)*.

– IETF RFC 1951 (1996), *DEFLATE Compressed Data Format Specification Version 1.3*. *(Spécification du format de données compressées DEFLATE version 1.3)*.

– IETF RFC 1991 (1996), *PGP Message Exchange Formats*. *(Formats d'échange de message PGP)*.

– IETF RFC 2045 (1996), *Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies*. *(Extensions de messagerie Internet multi-usage (MIME) Partie une: Format des messages Internet)*.

– IETF RFC 2119 (1997), *Keywords for use in RFCs to Indicate Requirement Levels*. *Mots clé à utiliser dans les RFC pour indiquer les niveaux d'exigence*.

– IETF RFC 2246 (1999), *The TLS Protocol Version 1.0*. *(Protocole TLS, version 1.0)*.

– IETF RFC 2253 (1997), *Lightweight Directory Access Protocol (v3): UTF-8 String Representation of Distinguished Names*. *Protocole allégé d'accès à un annuaire LDAP (v3): Représentation en chaîne de caractères UTF-8 des noms différenciés*.

– IETF RFC 2396 (1998), *Uniform Resource Identifiers (URI): Generic Syntax*. (*Identifiants de ressource uniformes (URI): Syntaxe générique*).

– IETF RFC 2535 (1999), *Domain Name System Security Extensions*. (*Extensions de sécurité au système de nom de domaine*).

– IETF RFC 2616 (1999), *Hypertext Transfer Protocol – HTTP/1.1*. (*Protocole de transfert hypertexte*).

– IETF RFC 2617 (1999), *HTTP Authentication: Basic and Digest Access Authentication*. (*Authentification HTTP: Authentification d'accès de base et par résumé*).

– IETF RFC 2798 (2000), *Definition of the inetOrgPerson LDAP Object Class*. (*Définition de la classe d'objet LDAP inetOrgPerson*).

– IETF RFC 2828 (2000), *Internet Security Glossary*. (*Glossaire de la sécurité Internet*).

– IETF RFC 2914 (2000), *Congestion Control Principles*. (*Principes de contrôle de l'encombrement*).

– IETF RFC 2915 (2000), *The Naming Authority Pointer (NAPTR) DNS Resource Record*. (*Enregistrement de ressource DNS de pointeur d'autorité de dénomination (NAPTR)*).

– IETF RFC 2945 (2000), *The SRP Authentication and Key Exchange System*. (*Système SRP d'authentification et d'échange de clé*).

– IETF RFC 2965 (2000), *HTTP State Management Mechanism*. (*Mécanisme de gestion d'état HTTP*).

– IETF RFC 3023 (2001), *XML Media Types*. (*Types de support XML*).

– IETF RFC 3061 (2001), *A URN Namespace of Object Identifiers*. (*Espace de nom d'URN d'identifiants d'objet*).

– IETF RFC 3075 (2001), *XML-Signature Syntax and Processing*. (*Syntaxe et traitement de signature XML*).

– IETF RFC 3377 (2002), *Lightweight Directory Access Protocol (v3): Technical Specification*. (*Protocole léger d'accès à un répertoire (v3): Spécification technique*).

– IETF RFC 3403 (2002), *Dynamic Delegation Discovery System (DDDS) Part Three: The Domain Name System (DNS) Database*. (*Système dynamique de découverte de délégation(DDDS) Partie trois: Base de données de système de nom de domaine (DNS)*).

– IETF RFC 3513 (2003), *Internet Protocol Version 6 (IPv6) Addressing Architecture*. *Architecture d'adressage du protocole Internet version 6 (IPv6).*

– IETF RFC 3546 (2003), *Transport Layer Security (TLS) Extensions*. *(Extensions à la sécurité de couche transport (TLS)*).

– IETF RFC 3923 (2004), *End-to-End Signing and Object Encryption for the Extensible Messaging and Presence Protocol (XMPP)*. (*Signature et chiffrement d'objet de bout en bout pour le protocole d'échange de message et de présence extensible (XMPP)*).

– IETF RFC 4122 (2005), *A Universally Unique IDentifier (UUID) URN Namespace*. (*Espace de nom d'URN d'identifiant mondialement unique* (UUID)).

– Liberty Alliance POAS:2003, R. Aarts, *Liberty Reverse HTTP Binding for SOAP Specification Version 1.0, Liberty Alliance Project*. (*Liaison HTTP inverse pour la spécification SOAP version 1.0*), Liberty Alliance Project.

– OASIS WSS:2006, WS-Security Core Specification 1.1. (*Spécification 1.1 centrale de la sécurité WS*).

– UNICODE-C, M. Davis; M. J. Dürst: *Unicode Normalization Forms.* UNICODE Consortium, mars 2001. (*Formes de normalisation Unicode*). UNICODE Consortium.

– W3C Canonicalization:2002, *Exclusive XML Canonicalization Version 1.0,* (*Canonisation XML exclusive, version 1.0*) W3C Recommendation, Copyright © [18 juillet 2002] World Wide Web Consortium, (Massachusetts Institute of Technology, *Institut National de Recherche en Informatique et en Automatique*, Keio University), http://www.w3.org/TR/xml-exc-c14n/.

– W3C Character Model:2005, *Character Model for the World Wide Web 1.0: Fundamentals*, (*Modèle de caractères pour la toile mondiale version 1.0*), W3C Recommendation, Copyright © [15 février 2005] World Wide Web Consortium, (Massachusetts Institute of Technology, *Institut National de Recherche en Informatique et en Automatique*, Keio University), Keio University), http://www.w3.org/TR/2005/REC-charmod-20050215/.

– W3C Datatypes:2001, XML Schema Part 2: Data types (*Schéma XML Partie 2: Types de données*), W3C Recommendation, Copyright © [2 mai 2001] World Wide Web Consortium, (Massachusetts Institute of Technology, *Institut National de Recherche en Informatique et en Automatique*, Keio University), http://www.w3.org/TR/2001/REC-xmlschema-2-20010502/.

– W3C Encryption:2002, *XML Encryption Syntax and Processing*, (*Syntaxe et traitement du chiffrement XML*), W3C Recommendation, Copyright © [10 décembre 2002] World Wide Web Consortium, (Massachusetts Institute of Technology, *Institut National de Recherche en Informatique et en Automatique*, Keio University, http://www.w3.org/TR/2002/REC-xmlenc-core-20021210/.

– W3C Web Services Glossary:2004, *Web Services Glossary*, (*Glossaire de services de la toile*), W3C Note, Copyright © [11 février 2004] World Wide Web Consortium, (Massachusetts Institute of Technology, *Institut National de Recherche en Informatique et en Automatique*, Keio University), http://www.w3.org/TR/ws-gloss/.

– W3C HTML:1999, *HTML 4.01 Specification*, (*Spécification de HTML version 4.01*), W3C Recommendation, Copyright © [24 décembre 1999] World Wide Web Consortium, (Massachusetts Institute of Technology, *Institut National de Recherche en Informatique et en Automatique*, Keio University), http://www.w3.org/TR/REC-html40/.

– W3C Namespaces:1999, *Namespaces in XML*, (*Espaces de nom en XML*), W3C Recommendation, Copyright © [14 janvier 1999] World Wide Web Consortium (Massachusetts Institute of Technology, *Institut National de Recherche en Informatique et en Automatique*, Keio University), http://www.w3.org/TR/REC-xml-names/.

– W3C Primer:2005, *SOAP Version 1.2 Part 0: Primer*, W3C Recommendation, Copyright © [24 juin 2005] World Wide Web Consortium, (Massachusetts Institute of Technology, *Institut National de Recherche en Informatique et en Automatique*, Keio University), http://www.w3.org/TR/2003/REC-soap12-part0-20030624/.

– W3C Signature:2002, *XML Signature Syntax and Processing*, (*Syntaxe et traitement de signature XML*), W3C Recommendation, Copyright © [12 février 2002] World Wide Web Consortium, (Massachusetts Institute of Technology, *Institut National de Recherche en Informatique et en Automatique*, Keio University), http://www.w3.org/TR/xmldsigcore/.

– W3C Signature Schema:2001, *XML Signature Schema*, (*Schéma de signature XML*), W3C Recommendation, Copyright © [1 mars 2001] World Wide Web Consortium, (Massachusetts Institute of Technology, *Institut National de Recherche en Informatique et en Automatique*, Keio University), http://www.w3.org/TR/xmldsig-core/xmldsig-core-schema.xsd.

– W3C String:1998, *Requirements for String Identity Matching and String Indexing*, (*Exigences pour la correspondance d'identité de chaînes et l'indexation de chaînes*), W3C Note, Copyright © [10 juillet 1998] World Wide Web Consortium, (Massachusetts Institute of Technology, *Institut National de Recherche en Informatique et en Automatique*, Keio University, http://www.w3.org/TR/WD-charreq.

– W3C SOAP:2000, *Simple Object Access Protocol* (*Protocole d'accès d'objet simple*) *(SOAP) 1.1*, W3C Note, Copyright © [8 mai 2000] World Wide Web Consortium, (Massachusetts Institute of Technology, *Institut National de Recherche en Informatique et en Automatique*, Keio University), http://www.w3.org/TR/2000/NOTE-SOAP-20000508.

– W3C XHTML:2002, *The Extensible HyperText Markup Language (Second Edition)*, (*Langage de balisage hyper-texte extensible (deuxième édition)*), W3C Recommendation, Copyright © [1 août 2002] World Wide Web Consortium, (Massachusetts Institute of Technology, *Institut National de Recherche en Informatique et en Automatique*, Keio University), http://www.w3.org/TR/xhtml1/.

– W3C XML 1.0:2004, *Extensible Markup Language (XML) 1.0 (Third Edition)*, (*Langage de balisage extensible (XML) 1.0 (troisième édition)*), W3C Recommendation, Copyright © [4 février 2004] World Wide Web Consortium, (Massachusetts Institute of Technology, *Institut National de Recherche en Informatique et en Automatique*, Keio University), http://www.w3.org/TR/REC-xml/.

– W3C XML Schema Part 1:2001, *XML Schema Part 1: Structures*, W3C Recommendation, Copyright © [2 mai 2001] World Wide Web Consortium, (Massachusetts Institute of Technology, *Institut National de Recherche en Informatique et en Automatique*, Keio University), http://www.w3.org/TR/2001/REC-xmlschema-1-20010502/.

NOTE – La référence à un document au sein de la présente Recommandation ne lui donne pas comme document autonome, le statut d'une Recommandation.

3 Définitions

Pour les besoins de la présente Recommandation, les définitions suivantes s'appliquent.

3.1 Définitions importées

3.1.1 La présente Recommandation utilise le terme suivant défini dans la Rec. UIT-T X.667:

a) UUID

3.1.2 La présente Recommandation utilise les termes suivants définis dans la Rec. UIT-T X.680:

a) identifiant d'objet;

b) notation de type ouvert.

3.1.3 La présente Recommandation utilise le terme suivant défini dans la Rec. UIT-T X.811:

a) principe.

3.1.4 La présente Recommandation utilise les termes suivants définis dans la Rec. UIT-T X.812:

a) information de contrôle d'accès;

b) utilisateur.

3.1.5 La présente Recommandation utilise les termes suivants définis dans le glossaire des services Web du W3C:

a) envoyeur SOAP initial;

b) espace de nom;

c) receveur SOAP ultime;

d) schéma XML.

3.1.6 La présente Recommandation utilise les termes suivants définis dans la RFC 2828 de l'IETF:

a) accès;

b) contrôle d'accès;

c) mandataire;

d) serveur mandataire;

e) tirer;

f) pousser;

g) architecture de sécurité;

h) politique de sécurité;

i) service de sécurité.

3.1.7 La présente Recommandation utilise les termes suivants définis dans la RFC 2396 de l'IETF:

a) identifiant de ressource uniforme (URI);

b) référence d'URI.

3.2 Définitions supplémentaires

3.2.1 **droits d'accès**: description du type d'interactions autorisées qu'un sujet peut avoir avec une ressource. Par exemple, lire, écrire, exécuter, ajouter, modifier, et supprimer.

3.2.2 **compte**: accord commercial formel pour la fourniture régulière de relations et services entre un principal et un fournisseur de services commerciaux.

3.2.3 **liaison de compte**: méthode de mise en relation de comptes chez deux fournisseurs différents qui représentent le même principal de sorte que les fournisseurs puissent communiquer au sujet du principal. Une liaison de compte peut être établie par le partage d'attributs ou par une fédération d'identité.

3.2.4 **rôle actif**: rôle qu'une entité système revêt en effectuant une opération, par exemple en accédant à une ressource.

3.2.5 **domaine administratif**: environnement ou contexte qui est défini par une combinaison d'une ou plusieurs politiques administratives, enregistrement de nom de domaine Internet, entités civiles légales (par exemple, individus, corporations, ou autres entités organisées de façon formelle), plus une collection d'hôtes, appareils de réseau et réseaux d'interconnexion (et d'autres traits possibles), plus des services et applications réseau (souvent variés) qui tournent sur eux. Un domaine administratif peut contenir ou définir un ou plusieurs domaines de sécurité. Un domaine administratif peut renfermer un seul site ou plusieurs. Les traits qui définissent un domaine administratif peuvent évoluer avec le temps, et le font dans de nombreux cas. Les domaines administratifs peuvent interagir et passer des accords pour fournir et/ou utiliser des services à travers les frontières des domaines administratifs.

3.2.6 **administrateur**: personne qui installe ou assure la maintenance d'un système ou qui l'utilise pour gérer des entités système, des utilisateurs, et/ou des contenus. Un administrateur est normalement affilié à un domaine administratif particulier et peut être affilié à plus d'un domaine administratif.

3.2.7 **affiliation, groupe d'affiliation**: ensemble d'entités système qui partagent un seul espace de nom (au sens fédéré) d'identifiants pour les principaux.

3.2.8 **anonymat**: qualité ou état de ce qui est anonyme, qui est la condition dans laquelle on a un nom ou une identité qui n'est pas connu ou est dissimulé.

3.2.9 **producteur d'assertions**: formellement, c'est le domaine administratif qui héberge une ou plusieurs autorités SAML. Informellement, une instance d'une autorité SAML.

3.2.10 **assertion**: ensemble de données produites par une autorité SAML concernant un acte d'authentification effectué sur un sujet, des informations d'attribut sur le sujet, ou des données d'autorisation qui s'appliquent au sujet par rapport à une ressource spécifiée.

3.2.11 **attribut**: caractéristique distincte d'un objet. Pour les objets réels, les attributs sont souvent spécifiés en termes de traits physiques, tels que taille, forme, poids, et couleur. Les objets dans le cyber-espace peuvent avoir des attributs qui décrivent la taille, le type de codage, l'adresse, le réseau, et ainsi de suite. Les attributs sont souvent représentés comme paires de "nom d'attribut" et "valeur d'attribut", par exemple, "foo" a la valeur 'bar', "count" a la valeur 1, "gizmo" a les valeurs "frob" et "2".

3.2.12 **assertion d'attribut**: assertion qui porte des informations sur les attributs d'un sujet.

3.2.13 **autorité d'attribut**: entité système qui produit des assertions d'attribut.

3.2.14 **authentification**: c'est le processus de détermination du fait que quelqu'un ou quelque chose est ce qu'il déclare être, avec un certain degré de confiance.

3.2.15 assertion d'authentification: assertion qui porte des informations sur un acte d'authentification réussi ayant eu lieu pour un sujet donné.

3.2.16 autorité d'authentification: entité système qui produit des assertions d'authentification.

3.2.17 autorisation: processus pour déterminer, par évaluation des informations de contrôle d'accès applicables, si un sujet a la permission d'avoir les types d'accès spécifiés à une ressource particulière. Ordinairement, l'autorisation est dans le contexte de l'authentification. Une fois qu'un sujet est authentifié, il peut être autorisé à effectuer différents types d'accès.

3.2.18 décision d'autorisation: résultat d'un acte d'autorisation. Le résultat peut être négatif, c'est-à-dire qu'il peut indiquer que le sujet n'a pas d'accès autorisé à la ressource.

3.2.19 assertion de décision d'autorisation: assertion qui porte des informations sur une décision d'autorisation.

3.2.20 canal de retour: canal de retour se réfère aux communications directes entre deux entités système sans message "redirecteur" à travers une autre entité système telle qu'un client HTTP (par exemple, un agent d'utilisateur).

3.2.21 liaison, liaison de protocole: de façon générique, spécification de la transposition, de façon concrète, de certains messages de protocole donnés, et peut-être de schémas d'échange de messages, dans un autre protocole. Par exemple, le mappage du message SAML `<AuthnRequest>` en HTTP est une liaison. Le mappage du même message SAML en SOAP est une autre liaison. Dans le contexte SAML, chaque liaison reçoit un nom dans le schéma "liaison SAML xxx".

3.2.22 accréditifs: données qui sont transférées pour établir une identité revendiquée de principal.

3.2.23 utilisateur final: personne naturelle qui fait usage de ressources pour les besoins d'une application.

3.2.24 entité: voir à entité système.

3.2.25 fédérer: établir une liaison ou lier ensemble deux entités ou plus.

3.2.26 fédération: ce terme est utilisé dans deux sens:

1) l'acte d'établir une relation entre deux entités.
2) une association comprenant tout nombre de fournisseurs de service et de fournisseurs d'identité.

3.2.27 identité fédérée: une identité de principal est dite fédérée entre un ensemble de fournisseurs lorsqu'il y a un accord entre les fournisseurs sur un ensemble d'identifiants et/ou d'attributs à utiliser pour se référer au principal.

3.2.28 canal frontal: canal frontal se réfère au "canal de communications" qui peut être réalisé entre deux serveurs parlant en HTTP en utilisant les messages "HTTP redirect" et donc qui se passent des messages l'un l'autre via un agent d'utilisateur, par exemple, un navigateur de la toile ou tout autre client HTTP.

3.2.29 identifiant: objet de données (par exemple, une chaîne) mappé sur une entité système qui se réfère de façon univoque à l'entité système. Une entité système peut avoir plusieurs identifiants distincts qui se réfèrent à elle. Un identifiant est par essence un "attribut distinctif" d'une entité.

3.2.30 identité: l'essence d'une entité. L'identité de quelqu'un est souvent décrite par ses caractéristiques, parmi lesquelles peuvent être un nombre quelconque d'identifiants.

3.2.31 défédération d'identité: action qui survient lorsque les fournisseurs s'accordent pour cesser de se référer à un principal via un certain ensemble d'identifiants et/ou d'attributs.

3.2.32 fédération d'identité: acte de création d'une identité fédérée au nom d'un principal.

3.2.33 fournisseur d'identité: sorte de fournisseur de service qui crée, assure la maintenance, et gère les informations d'identité pour les principaux et fournit l'authentification du principal aux autres fournisseurs de service au sein d'une fédération, comme avec les profils de navigateur de la toile.

3.2.34 fournisseur d'identité léger: sorte de fournisseur de service qui crée, assure la maintenance, et gère les informations d'identité pour les principaux et fournit l'authentification du principal aux autres fournisseurs de service au sein d'une fédération, en utilisant seulement les portions nécessaires de SAML.

3.2.35 ouverture de session, inscription: processus par lequel un utilisateur présente des accréditifs à une autorité d'authentification, établit une session simple, et facultativement, établit une session riche.

3.2.36 fermeture de session, désincription: processus par lequel un utilisateur signifie son désir de terminer une session simple ou riche.

3.2.37 langage de balisage: ensemble d'éléments XML et d'attributs XML à appliquer à la structure d'un document XML pour un besoin spécifique. Un langage de balisage est normalement défini au moyen d'un ensemble de schémas XML et de la documentation d'accompagnement.

3.2.38 qualificatif de nom: chaîne qui précise de façon non ambiguë un identifiant qui peut être utilisé dans plus d'un espace de nom (au sens fédéré) pour représenter des principaux différents.

3.2.39 partie: de façon informelle, un ou plusieurs principaux participant à un processus ou une communication, comme de recevoir une assertion ou d'accéder à une ressource.

3.2.40 pseudonyme permanent: identifiant de nom préservant la confidentialité qui est alloué par un fournisseur pour identifier un principal auprès d'une partie support donnée pour une longue période qui s'étend sur plusieurs sessions; il peut être utilisé pour représenter une fédération d'identité.

3.2.41 point de décision de politique (PDP): entité système qui prend des décisions d'autorisation pour elle-même ou pour d'autres entités système qui demandent de telles décisions. Par exemple, un PDP SAML consomme des demandes de décision d'autorisation, et produit en réponse des assertions de décision d'autorisation. Un PDP est une "autorité de décision d'autorisation".

3.2.42 point de mise en application de politique (PEP, *policy enforcement point*): entité système qui demande et ensuite met en application des décisions d'autorisation. Par exemple, un PEP SAML envoie des demandes de décision d'autorisation à un PDP, et consomme les assertions de décision d'autorisation envoyées en réponse.

3.2.43 identité principale: représentation de l'identité d'un principal, normalement un identifiant.

3.2.44 profil: ensemble de règles pour un ou plusieurs besoins; chaque ensemble est doté d'un nom dans le schéma "profil xxx de SAML" ou "profil SAML xxx".

1) règles sur la manière d'enchasser des assertions dans un protocole ou autre contexte d'usage et de les en extraire;
2) règles d'utilisation des messages du protocole SAML dans un contexte d'utilisation particulier;
3) règles de mappage des attributs exprimés dans SAML dans d'autres systèmes de représentation d'attributs. Un tel ensemble de règles est appelé "profil d'attribut".

3.2.45 liaison de protocole: (voir § 3.2.20).

3.2.46 fournisseur: façon générique de désigner à la fois les fournisseurs d'identité et les fournisseurs de service.

3.2.47 partie support: entité système qui décide d'entreprendre une action sur la base des informations provenant d'une autre entité système. Par exemple, une partie support SAML dépend de la réception d'assertions provenant d'un producteur d'assertions (une autorité SAML) sur un sujet.

3.2.48 femandeur: entité système qui utilise le protocole SAML pour demander des services de la part d'une autre entité système (une autorité SAML, un répondant). Le terme de "client" n'est pas utilisé pour cette notion parce que de nombreuses entités système agissent simultanément ou en série à la fois comme clients et comme serveurs. Dans les cas où on utilise la liaison SOAP pour SAML, la demande SAML est architecturalement distincte de l'envoyeur SOAP initial.

3.2.49 ressource: données contenues dans un système d'informations (par exemple, sous forme de fichiers, d'informations en mémoire, etc), ainsi que:

1) un service fourni par un système;
2) un élément d'équipement système (autrement dit, un composant de système tel qu'un matériel, un microcode, un logiciel ou une documentation).

3.2.50 répondant: une entité système (une autorité SAML) qui utilise le protocole SAML pour répondre à une demande de services de la part d'une autre entité système (un demandeur). Le terme de "serveur" n'est pas utilisé pour cette notion parce que de nombreuses entités système agissent simultanément ou en série à la fois comme clients et serveurs. Dans les cas où la liaison SOAP est utilisée pour SAML, le répondant SAML est architecturalement distinct du receveur SOAP ultime.

3.2.51 rôle: les dictionnaires définissent un rôle comme "un caractère ou un texte joué par un acteur" ou "une fonction ou position." Les entités système jouent divers types de rôles de façon sérielle et/ou simultanément, par exemple, des rôles actifs et des rôles passifs. La notion d'administrateur est souvent un exemple de rôle.

3.2.52 artifice SAML: petit objet de données structurées de taille fixe pointant sur un message de protocole SAML de taille variable, normalement plus grand. Les artifices SAML sont conçus pour être enchassés dans les URL et transportés dans les messages HTTP, comme les messages de réponse HTTP avec les codes d'état "Redirection 3xx", et les messages GET HTTP suivants. De cette façon, un fournisseur de service peut porter indirectement, via un agent

d'utilisateur, un artifice SAML à un autre fournisseur, qui peut ultérieurement déréférencer l'artifice SAML via une interaction directe avec le fournisseur effectif, et obtenir le message de protocole SAML.

3.2.53 autorité SAML: entité système abstraite dans le modèle de domaine SAML qui produit les assertions. Voir aussi à autorité d'attribut, autorité d'authentification, et point de décision de politique (PDP).

3.2.54 sécurité: collection de sauvegardes qui assurent la confidentialité de l'information, protègent les systèmes ou réseaux utilisés pour les traiter, et contrôler leur accès. La sécurité englobe normalement les concepts de secret, de confidentialité, d'intégrité et de disponibilité. Elle est destinée à garantir qu'un système a un potentiel de résistance à des attaques.

3.2.55 assertion de sécurité: assertion qui est examinée dans le contexte d'une architecture de sécurité.

3.2.56 contexte de sécurité: par rapport à un message individuel de protocole SAML, le contexte de sécurité de message est l'union sémantique des blocs d'en-tête de sécurité (s'il en est) du message avec les autres mécanismes de sécurité qui peuvent être employés dans la livraison du message à un receveur. Par rapport à ce dernier, des exemples de mécanisme de sécurité employé dans les couches inférieures de la pile réseau sont HTTP, TLS et IPSec.

3.2.57 domaine de sécurité: environnement ou contexte défini par les modèles de sécurité et l'architecture de sécurité, y compris un ensemble de ressources et un ensemble d'entités système qui sont autorisées à accéder aux ressources. Un ou plusieurs domaines de sécurité peuvent résider dans un seul domaine administratif. Les traits qui définissent un domaine de sécurité donné évoluent normalement dans le temps.

3.2.58 expression de politique de sécurité: transposition des identités principales et/ou des attributs en actions admissibles. Les expressions de politique de sécurité sont souvent essentiellement des listes de contrôle d'accès.

3.2.59 fournisseur de service: rôle joué par une entité système où l'entité système fournit des services à des entités système principales ou autres.

3.2.60 fournisseur de service léger: rôle joué par une entité système où l'entité système fournit des services à des entités système principales ou autres en utilisant seulement la portion nécessaire du protocole SAML.

3.2.61 session: interaction durable entre des entités système, impliquant souvent un Principal, caractérisé par le maintien d'un certain état de l'interaction pour la durée de l'interaction.

3.2.62 autorité de session: rôle joué par une entité système pendant qu'elle maintient l'état relatif aux sessions.

3.2.63 participant de session: rôle joué par une entité système lorsqu'elle participe à une session avec au moins une autorité de session.

3.2.64 désinscription: voir "fermeture de session".

3.2.65 inscription: voir "ouverture de session".

3.2.66 site: terme informel pour désigner un domaine administratif au sens géographique ou de nom DNS. Il peut se référer à une portion géographique ou topologique particulière d'un domaine administratif, ou il peut recouvrir plusieurs domaines administratifs, comme cela peut être le cas sur un site ASP.

3.2.67 sujet: un principal dans le contexte d'un domaine de sécurité. Les assertions SAML font des déclarations sur les sujets.

3.2.68 entité système, entité: élément actif d'un système ordinateur/réseau. Par exemple, un processus automatisé ou un ensemble de processus, un sous-système, une personne ou groupe de personnes qui incorpore un ensemble distinct de fonctionnalités.

3.2.69 fin de temporisation: période après laquelle certaines conditions deviennent vraies si certains événements ne sont pas survenus. Par exemple, une session qui s'est terminée parce que son état a été inactif pendant une période spécifiée est dite "arrivée en fin de temporisation".

3.2.70 pseudonyme transitoire: identifiant de préservation de la confidentialité alloué par un fournisseur d'identité pour identifier un principal auprès d'une partie support donnée pour une période relativement brève qui n'a pas besoin de s'étendre sur plusieurs sessions.

3.2.71 attribut XML: structure de données XML qui est enchassée dans l'étiquette de début d'un élément XML et qui a un nom et une valeur.

3.2.72 élément XML: structure de données XML qui est arrangée hiérarchiquement parmi d'autres, telles que des structures dans un document XML, et qui est indiquée par une étiquette de début et une étiquette de fin ou une étiquette vide.

4 Abréviations

Pour les besoins de la présente Recommandation, les abréviations suivantes s'appliquent:

AA	autorité d'attribut (*attribute authority*)
ASN.1	notation de syntaxe abstraite n° 1 (*abstract syntax notation one*)
ASP	fournisseur de service d'application (*application service provider*)
CA	autorité de certification (*certification authority*)
CMP	protocole de gestion de certificats (*certificate management protocol*)
CRL	liste de révocation de certificat (*certificate revocation list*)
DCE	environnement de calcul distribué (*distributed computing environment*)
DDDS	système dynamique de découverte de délégations (*dynamic d elegation discovery system*)
DNS	système de dénomination de domaine (*domain name system*)
ECP	client/mandataire amélioré (*enhanced client/proxy*)
HTTP	protocole de transfert hypertexte (*hypertext transfer protocol*)
HTTPS	protocole sécurisé de transport hypertexte (*secure hypertext transport protocol*)
IdP	fournisseur d'identité (*Identity provider*)
IdP Lite	fournisseur d'identification légère (*identity provider lite*)
IP	protocole Internet (*Internet protocol*)
IPSec	sécurité du protocole Internet (*Internet protocol security*)
MD5	algorithme 5 de résumé de message (*message digest algorithm 5*)
MIME	extensions multi-usage de messagerie Internet (*multipurpose internet mail extensions*)
NAPTR	pointeur d'autorité de nommage (*naming authority pointer*)
OID	identificateur d'objet (*object identifier*)
PAC	certificats d'attribut privilégié (*privilege attribute certificates*)
PAOS	SOAP inversé (*reverse SOAP*)
PDP	point de décision de politique (*policy decision point*)
PEP	point de mise en application de politique (*policy enforcement point*)
PGP	bonne confidentialité (*pretty good privacy*)
PKI	infrastructure de clé publique (*public-key infrastructure*)
POP	preuve de possession (*proof of possession*)
RA	autorité d'enregistrement (*registration authority*)
RSA	algorithme à clé publique de Rivest Shamir Adleman (*Rivest Shamir Adleman*) (*public key algorithm*)
SHA-1	algorithme de hachage sécurisé n° 1 (*secure hash algorithm 1*)
SP	fournisseur de service (*service provider*)
SPKI	infrastructure simple de clé publique (*simple public key infrastructure*)
SP Lite	fournisseur de service léger (*service provider lite*)
SSO	inscription unique (*single sign on*)
TLS	protocole de sécurité de la couche transport (*transport layer security protocol*)
URI	identificateur de ressource uniforme (*uniform resource identifier*)
UTC	temps universel coordonné (*coordinated universal time*)
UUID	identificateur unique universel (*universal unique identifier*)
XACML	langage extensible de balisage de contrôle d'accès (*extensible access control markup language*)
XML	langage de balisage extensible (*extensible markup language*)

5 Conventions

La présente Recommandation utilise les mots clé "doit", "ne doit pas", "exige", "devra", "ne devra pas", "devrait", "ne devrait pas", "recommande", "peut", et "facultatif". Dans la présente Recommandation, ces termes sont à interpréter comme décrit dans la RFC 2119 de l'IETF.

La présente Recommandation utilise les documents de schéma XML se conformant à la partie 1 du schéma XML du W3C, à la partie 2 du schéma XML du W3C et au texte normatif de ces spécifications qui décrivent la syntaxe et la sémantique des assertions et messages de protocole SAML codés en XML. En cas de désaccord entre les documents de schéma SAML et les listes de schéma de la présente Recommandation, les documents de schéma prennent le pas. Noter que dans certains cas, la présente Recommandation impose des contraintes qui vont au-delà de celles indiquées par les documents de schéma.

6 Aperçu général

La présente Recommandation est destinée à spécifier la version 2 du langage de balisage d'assertions de sécurité (SAML). Elle définit la syntaxe et la sémantique de traitement des assertions faites à propos d'un sujet par une entité système. Tout en faisant de telles assertions, ou en s'appuyant sur elles, les entités système SAML peuvent utiliser d'autres protocoles pour communiquer sur une assertion elle-même, ou sur le sujet d'une assertion. La présente Recommandation définit la structure des assertions SAML, l'ensemble des protocoles associés, et les règles de traitement impliquées par la gestion d'un système SAML.

Les assertions et les messages de protocole SAML sont codés en XML et utilisent les espaces de nom XML. Ils sont normalement enchassés dans d'autres structures pour le transport, comme les demandes POST de HTTP ou les messages SOAP codés en XML. Le paragraphe 7 définit les types de données communs aux utilisations de SAML. Le paragraphe 8 donne un cadre de travail pour les assertions et les protocoles SAML. Le paragraphe 9 décrit le modèle de métadonnées SAML. Le paragraphe 10 développe les cadres pour l'enchassement et le transport des messages de protocole SAML. Le paragraphe 11 donne un ensemble de base de profils à l'usage des assertions et protocoles SAML pour des cas d'utilisation particuliers ou pour réaliser l'interopérabilité lors de l'utilisation de caractéristiques SAML. Le paragraphe 12 expose le contexte d'authentification de SAML. En particulier,les contextes suivants sont spécifiés:

- schéma de contexte d'authentification SAML;
- types de schéma de contexte d'authentification SAML;
- schéma de classe de contexte SAML pour protocole Internet;
- schéma de classe de contexte SAML pour mot de passe de protocole Internet;
- schéma de classe de contexte SAML pour Kerberos;
- schéma de classe de contexte SAML pour mobile de facteur non enregistré;
- schéma de classe de contexte SAML pour mobile de facteur deux non enregistré;
- schéma de classe de contexte SAML pour mobile de facteur un à contrat;
- schéma de classe de contexte SAML pour mobile de facteur deux à contrat;
- schéma de classe de contexte SAML pour mot de passe;
- schéma de classe de contexte SAML pour transport protégé par mot de passe;
- schéma de classe de contexte SAML pour session antérieure;
- schéma de classe de contexte SAML pour clé publique – X.509;
- schéma de classe de contexte SAML pour clé publique – PGP;
- schéma de classe de contexte SAML pour clé publique – SPKI;
- schéma de classe de contexte SAML pour clé publique – signature XML;
- schéma de classe de contexte SAML pour carte à mémoire;
- schéma de classe de contexte SAML pour PKI à carte à mémoire;
- schéma de classe de contexte SAML pour PKI logiciel;
- schéma de classe de contexte SAML pour la téléphonie;
- schéma de classe de contexte SAML pour la téléphonie (nomade);
- schéma de classe de contexte SAML pour la téléphonie (personalisée);
- schéma de classe de contexte SAML pour la téléphonie (authentifiée);
- schéma de classe de contexte SAML pour mot de passe distant sécurisé;

- schéma de classe de contexte SAML pour authentification client fondée sur certificat SL/TLS;
- schéma de classe de contexte SAML pour jeton synchronisé.

Le paragraphe 13 procure aux développeurs de SAML un cadre de travail qui doit être suivi pour assurer la conformité. Au paragraphe 13, les exigences de conformité sont exposées avec les modes de fonctionnement et les modèles de sécurité. L'Annexe A comporte une liste de tous les schémas SAML associés.

7 Types de données communs

Les paragraphes qui suivent définissent l'utilisation et l'interprétation des types de données communs qui apparaissent dans les schémas SAML.

7.1 Valeurs de chaînes

Toutes les valeurs de chaînes de SAML ont le type **xs:string**, qui est construit en datatypes XML du W3C. Sauf notation contraire dans la présente Recommandation, toutes les chaînes dans les messages SAML doivent consister en au moins un caractère qui ne soit pas un espace blanc.

Sauf notation contraire dans la présente Recommandation ou un profil particulier, tous les éléments dans les documents SAML qui ont le schéma XML **xs:string** type, ou un type qui en est dérivé, doivent être comparés en utilisant une comparaison binaire exacte. En particulier, les implémentations et développements de SAML ne doivent pas dépendre de comparaisons de chaînes insensibles à la casse, de normalisation ou d'élagage d'espaces blancs ou de conversion de formats locaux spécifiques tels que des nombres ou des monnaies. Cette exigences est destinée à se conformer à la Chaîne W3C.

Si une implémentation compare des valeurs qui sont représentées en utilisant différents codages de caractères, elle doit utiliser une méthode de comparaison qui retourne le même résultat que par la conversion des deux valeurs en codage de caractère Unicode, forme de normalisation C, et en effectuant ensuite une comparaison binaire exacte. Cette exigence est destinée à se conformer au modèle de caractère W3C et en particulier aux règles pour le texte normalisé en Unicode.

Les applications qui comparent les données reçues dans les documents SAML à des données provenant de sources externes doivent tenir compte des règles de normalisation spécifiées pour XML. Le texte contenu dans les éléments est normalisé de telle sorte que les fins de ligne soient représentées en utilisant les caractères de renvoi à la ligne (ASCII CODE $10_{Decimal}$). Les valeurs d'attribut XML définies comme chaînes (ou types dérivés de chaînes) sont normalisées comme décrit au paragraphe 3.3.3 de XML 1.0 du W3C. Tous les caractères espace blanc sont remplacés par des blancs (ASCII CODE $32_{Decimal}$).

La présente Recommandation ne définit pas d'ordre de collationnement ou de tri des valeurs d'attribut XML ou de contenu d'élément. Les implémentations de SAML ne doivent pas dépendre d'un ordre de tri spécifique pour les valeurs, parce qu'elles peuvent différer selon les réglages locaux des hôtes impliqués.

7.2 Valeurs d'URI

Toutes les valeurs de référence d'URI de SAML ont le type **xs:anyURI**, qui est construit en datatypes XML du W3C.

Sauf notation contraire dans la présente Recommandation, toutes les valeurs de référence d'URI au sein d'éléments ou attributs définis en SAML doivent consister en au moins un caractère qui ne soit pas un espace blanc, et dont il est exigé qu'il soit absolu.

La présente Recommandation utilise de façon étendue les références d'URI comme identifiants, tels que des codes d'état, des types de format, des noms d'attribut et d'entité système, etc. Donc, il est essentiel que les valeurs soient à la fois uniques et cohérentes, de sorte que le même URI ne soit jamais utilisé à différents moments pour représenter des informations sous-jacentes différentes.

7.3 Valeurs horaires

Toutes les valeurs horaires de SAML ont le type **xs:dateTime**, qui est construit en datatypes XML du W3C et doit être exprimé dans la forme UTC, sans composant de fuseau horaire.

Les entités système SAML ne devraient pas s'appuyer sur une résolution horaire plus fine que la milliseconde. Les implémentations ne doivent pas générer d'instants horaires qui spécifient les secondes sautées.

7.4 ID et valeurs de référence d'ID

Le type simple **xs:ID** est utilisé pour déclarer les identifiants SAML pour les assertions, demandes, et réponses. Les valeurs déclarées comme étant du type **xs:ID** dans la présente Recommandation doivent satisfaire aux propriétés suivantes en plus de celles imposées par la définition du type **xs:ID** lui-même:

- toute partie qui alloue un identifiant doit s'assurer qu'il y a une probabilité négligeable que cette partie ou toute autre partie alloue accidentellement le même identifiant à un objet de données différent;
- lorsqu'un objet de données déclare qu'il a un identifiant particulier, il ne doit y avoir exactement qu'une seule telle déclaration.

Le mécanisme par lequel une entité système SAML s'assure de l'unicité de l'identifiant appartient à l'implémentation. Dans le cas d'utilisation d'une technique aléatoire ou pseudo-aléatoire, la probabilité que deux identifiants choisis de façon aléatoire soient identiques doit être inférieure ou égale à 2^{-128} et ne devrait pas être inférieure ou égale à 2^{-160}. Cette exigence peut être satisfaite en codant une valeur choisie aléatoirement dans une longueur binaire comprise entre 128 et 160 bits. Le codage doit se conformer aux règles qui définissent le datatype **xs:ID**. Un générateur pseudo-aléatoire doit être alimenté avec du matériel unique afin d'assurer les propriétés d'unicité désirées entre les différents systèmes.

Le type simple **xs:NCName** est utilisé en SAML pour référencer des identifiants de type **xs:ID** car **xs:IDREF** ne peut pas être utilisé à cette fin. En SAML, l'élément désigné par une référence d'identifiant SAML pourrait en fait être défini dans un document distinct de celui dans lequel est utilisée la référence d'identifiant. Utiliser **xs:IDREF** violerait l'exigence que cette valeur corresponde à la valeur d'un attribut d'ID sur un élément quelconque du même document XML.

8 Assertions et protocoles SAML

SAML définit la syntaxe et la sémantique de traitement des assertions faites sur un sujet par une entité système. En utilisant ou en s'appuyant sur de telles assertions, les entités système SAML peuvent utiliser d'autres protocoles pour communiquer une assertion elle-même, ou le sujet d'une assertion. Le présent paragraphe définit la structure des assertions SAML, un ensemble de protocoles associés, ainsi que les règles de traitement impliquées dans la gestion d'un système SAML.

Les assertions SAML et les messages de protocole sont codés en XML (voir XML 1.0 du W3C) et utilisent les espaces de nom XML (voir Namespaces du W3C). Ils sont normalement enchassés dans d'autres structures pour le transport, telles que les demandes POST de HTTP ou les messages SOAP codés en XML. Le paragraphe 10 fournit un cadre de travail pour l'enchassement et le transport des messages de protocole SAML. Le paragraphe 11 donne un ensemble de base de profils pour l'utilisation des assertions et protocoles SAML pour répondre à des cas d'usage spécifiques ou réaliser l'interopérabilité lors de l'utilisation de caractéristiques SAML.

8.1 Assertions SAML

Une assertion est un paquetage d'informations qui fournit zéro ou plusieurs déclarations faites par une **autorité SAML**; les autorités SAML sont parfois mentionnées comme **producteurs d'assertions** dans les discussions sur la génération et l'échange d'assertions, et les entités système qui utilisent les assertions reçues sont appelées **consommateurs d'assertions**. (Ces termes sont différents de **demandeur** et **receveur**, qui sont réservés aux discussions d'échange de messages de protocole SAML.)

Les assertions SAML sont habituellement faites à propos d'un **sujet**, représenté par l'élément `<Subject>`. Cependant, l'élément `<Subject>` est facultatif, et d'autres spécifications et profils peuvent utiliser la structure d'assertion SAML pour faire des déclarations similaires sans spécifier un sujet, ou en spécifiant le sujet d'une autre façon. Il y a normalement un certain nombre de **fournisseurs de service** qui peuvent utiliser des assertions sur un sujet afin de contrôler l'accès et fournir un service personnalisé, et en conséquence, ils deviennent les consommateurs d'assertions d'un producteur d'assertions appelé **fournisseur d'identité**.

La présente Recommandation définit trois différentes sortes de déclarations d'assertion qui peuvent être créées par une autorité SAML. Toutes les déclarations définies comme SAML sont associées à un sujet. Les trois sortes de déclaration définies dans la présente Recommandation sont:

- **authentification**: le sujet de l'assertion a été authentifié par un moyen particulier à un moment précis;
- **attribut**: le sujet de l'assertion est associé aux attributs fournis;
- **décision d'autorisation**: une demande d'autorisation du sujet de l'assertion à accéder aux ressources spécifiées a été accordée ou refusée.

NOTE (informative) – PE13 (voir OASIS PE:2006) suggère d'ajouter "ou est indéterminée" à l'alinéa ci-dessus.

La structure externe d'une assertion est générique, fournissant des informations qui sont communes à toutes les déclarations qu'elle contient. Au sein d'une assertion, une série d'éléments internes décrit l'authentification, l'attribut, la décision d'autorisation, ou les déclarations définies par l'utilisateur qui contiennent les éléments spécifiques.

Les extensions sont permises par le schéma d'assertions SAML, comme décrit au paragraphe 8.6, permettant des extensions définies par l'utilisateur aux assertions et déclarations, et permettant la définition de nouvelles sortes d'assertions et déclarations.

8.1.1 En-tête de schéma et déclarations d'espace de nom

Le fragment de schéma suivant définit les espaces de nom XML et les autres informations d'en-tête pour le schéma d'assertion:

```
<schema targetNamespace="urn:oasis:names:tc:SAML:2.0:assertion"
    xmlns="http://www.w3.org/2001/XMLSchema"
    xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion"
    xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
    xmlns:xenc="http://www.w3.org/2001/04/xmlenc#"
    elementFormDefault="unqualified"
    attributeFormDefault="unqualified"
    blockDefault="substitution"
    version="2.0">
    <import namespace="http://www.w3.org/2000/09/xmldsig#"
        schemaLocation="http://www.w3.org/TR/2002/REC-xmldsig-core-
20020212/xmldsig-core-schema.xsd"/>
    <import namespace="http://www.w3.org/2001/04/xmlenc#"
        schemaLocation="http://www.w3.org/TR/2002/REC-xmlenc-core-
20021210/xenc-schema.xsd"/>
    <annotation>
        <documentation>
            Document identifier: saml-schema-assertion-2.0
            Location: http://docs.oasis-open.org/security/saml/v2.0/
            Revision history:
            V1.0 (November, 2002):
              Initial Standard Schema.
            V1.1 (September, 2003):
              Updates within the same V1.0 namespace.
            V2.0 (March, 2005):
              New assertion schema for SAML V2.0 namespace.
        </documentation>
    </annotation>
…
</schema>
```

8.1.2 Identifiants de nom

Les paragraphes qui suivent définissent la construction SAML qui contient les identifiants descriptifs pour les sujets et les producteurs d'assertions et les messages de protocole.

Il y a un certain nombre de circonstances qui sont utiles dans SAML pour que deux entités système communiquent au sujet d'une tierce partie; par exemple, le protocole de demande d'authentification SAML permet l'authentification d'un sujet par une tierce partie. Et donc, il est utile d'établir un moyen par lequel les parties puissent être associées à des identifiants significatifs pour chacune des parties. Dans certains cas, il sera nécessaire de limiter le domaine au sein duquel un identifiant est utilisé à un petit ensemble d'entités système (pour préserver la confidentialité d'un sujet, par exemple). Des identifiants similaires peuvent aussi être utilisés pour se référer au producteur d'un message de protocole ou assertion SAML.

Il est possible que deux entités système ou plus utilisent la même valeur d'identifiant de nom lorsqu'elles se réfèrent à des identités différentes. Et donc, chaque entité peut avoir une compréhension différente de ce même nom. SAML fournit des **qualificatifs de noms** pour ôter toute ambiguïté sur un identifiant de nom en le plaçant effectivement dans un **espace de nom** ***(namespace)*** fédéré se rapportant aux qualificatifs de nom. SAML v2.0 permet à un identifant d'être qualifié en termes à la fois de producteur d'assertions et de consommateur d'assertions ou d'affiliation particulier, permettant aux identifiants d'exhiber, quand nécessaire, une sémantique d'appariement.

Les identifiants de nom peuvent aussi être chiffrés pour améliorer encore leurs caractéristiques de préservation de la confidentialité, en particulier dans les cas où l'identifiant va être transmis via un intermédiaire.

NOTE – Pour éviter l'utilisation des constructions de schéma XLM relativement complexes, les divers types d'éléments identifiants ne partagent pas une hiérarchie de type commune.

8.1.2.1 Élément <BaseID>

L'élément `<BaseID>` est un point d'extension qui permet aux applications d'ajouter de nouvelles sortes d'identifiants. Son type complexe **BaseIDAbstractType** est abstrait et n'est donc utilisable que comme base d'un type déduit. Il inclut les attributs suivants à utiliser par les représentations d'identifiant étendues:

– `NameQualifier` [Facultatif]

Domaine de sécurité ou administratif qui qualifie l'identifiant. Cet attribut donne le moyen de fédérer des identifiants provenant de mémoires d'utilisateurs disparates sans collision.

– `SPNameQualifier` [Facultatif]

Ajoute à la qualification d'un identifiant avec le nom d'un fournisseur de service ou une affiliation de fournisseurs. Cet attribut fournit un moyen supplémentaire de fédérer des identifiants sur la base du ou des consommateurs d'assertions.

Les attributs `NameQualifier` et `SPNameQualifier` devraient être omis sauf si la définition du type d'identifiant définit explicitement leur utilisation et leur sémantique.

Le fragment de schéma suivant définit l'élément `<BaseID>` et son type complexe **BaseIDAbstractType**:

```
<attributeGroup name="IDNameQualifiers">
    <attribute name="NameQualifier" type="string" use="optional"/>
    <attribute name="SPNameQualifier" type="string" use="optional"/>
</attributeGroup>
<element name="BaseID" type="saml:BaseIDAbstractType"/>
<complexType name="BaseIDAbstractType" abstract="true">
    <attributeGroup ref="saml:IDNameQualifiers"/>
</complexType>
```

8.1.2.2 Type complexe NameIDType

Le type complexe **NameIDType** est utilisé lorsqu'un élément sert à représenter une entité par un nom valorisé par une chaîne. C'est une forme d'identifiant plus restreinte que l'élément `<BaseID>` et c'est le type sous-jacent à la fois à l'élément `<NameID>` et à l'élément `<Issuer>`. En plus du contenu de la chaîne qui détient l'identifiant réel, il fournit les attributs facultatifs suivants:

– `NameQualifier` [Facultatif]

Domaine de sécurité ou administratif qui qualifie le nom. Cet attribut donne le moyen de fédérer les noms provenant de mémoires d'utilisateur disparates sans collision.

– `SPNameQualifier` [Facultatif]

Ajoute à la qualification d'un nom avec le nom d'un fournisseur de service ou d'affiliation de fournisseurs. Cet attribut fournit un moyen supplémentaire de fédérer des noms sur la base du ou des consommateurs d'assertions.

– `Format` [Facultatif]

Référence d'URI qui représente la classification des informations d'identifiant fondées sur une chaîne. Voir au paragraphe 8.7.3 les définitions SAML de référence d'URI qui peuvent être utilisées comme valeurs de l'attribut Format et leurs descriptions et règles de traitement associées. Sauf spécification contraire par un élément sur la base de ce type, si aucune valeur `Format` n'est fournie, la valeur `urn:oasis:names:tc:SAML:1.0:nameid-format:unspecified` (voir au § 8.7.3.1) est en vigueur.

Lorsqu'une valeur `Format` autre que celle spécifiée au § 8.7.3 est utilisée, le contenu d'un élément de ce type doit être interprété conformément à la définition de ce format fournie en-dehors de la présente Recommandation. Sauf indication contraire par la définition du format, les questions d'anonymat, de pseudonyme, de persistence de l'identifiant par rapport aux producteurs et consommateurs d'assertions relèvent de l'implémentation.

– `SPProvidedID` [Facultatif]

Identifiant de nom établit par un fournisseur de service ou une affiliation de fournisseurs pour l'entité, s'il diffère de l'identifiant de nom principal donné dans le contenu de l'élément. Cet attribut donne le moyen d'intégrer l'utilisation de SAML avec des identifiants existants déjà utilisés par un fournisseur de service. Par exemple, un identifiant existant peut être "attaché" à l'entité en utilisant le protocole de gestion d'identifiant de nom défini au § 8.2.8.

Des règles supplémentaires pour le contenu de ces attributs (ou leur omission) peuvent être définies par des éléments qui utilisent ce type, et par des définitions de `Format` spécifiques. Les attributs `NameQualifier` et `SPNameQualifier` devraient être omis sauf si l'élément ou le format définit explicitement leur usage et leur sémantique.

Le fragment de schéma suivant définit le type complexe **NameIDType**:

```
<complexType name="NameIDType">
    <simpleContent>
         <extension base="string">
              <attributeGroup ref="saml:IDNameQualifiers"/>
              <attribute name="Format" type="anyURI" use="optional"/>
              <attribute name="SPProvidedID" type="string" use="optional"/>
         </extension>
    </simpleContent>
</complexType>
```

8.1.2.3 Élément <NameID>

L'élement `<NameID>` est du type **NameIDType** (voir au § 8.1.2.2), et il est utilisé dans diverses construtions d'assertions SAML telles que les éléments `<Subject>` et `<SubjectConfirmation>`, et dans divers messages de protocole (voir au § 8.2).

Le fragment de schéma suivant définit l'élément `<NameID>`:

```
<element name="NameID" type="saml:NameIDType"/>
```

8.1.2.4 Élément <EncryptedID>

L'élément `<EncryptedID>` est du type **EncryptedElementType**, et porte le contenu d'un élément d'identifiant non chiffré en mode chiffré, comme défini par Chiffrement du W3C. L'élément `<EncryptedID>` contient les éléments suivants:

– `<xenc:EncryptedData>` [Exigé]

  Contenu chiffré et détails de chiffrement associés, comme défini par Chiffrement du W3C. L'attribut Type devrait être présent et, s'il est présent, doit contenir une valeur de `http://www.w3.org/2001/04/xmlenc#Element`. Le contenu chiffré doit contenir un élément du type **NameIDType** ou **AssertionType**, ou un type déduit de **BaseIDAbstractType**, **NameIDType**, ou **AssertionType**.

– `<xenc:EncryptedKey>` [Zero or More]

  Clés de déchiffrement enveloppées comme définit dans Chiffrement du W3C. Chaque clé enveloppée devrait inclure un attribut Recipient qui spécifie l'entité pour laquelle la clé a été chiffrée. La valeur de l'attribut Recipient devrait être l'identifiant d'URI d'une entité système SAML, comme définie au § 8.4.

Les identifiants chiffrés sont destinés à jouer le rôle de mécanisme de protection de la confidentialité lorsque la valeur en texte clair passe à travers un intermédiaire. Comme tel, le texte chiffré doit être unique pour toute opération de chiffrement donnée. Pour toute précision sur cette question, voir au § 6.3 de Chiffrement XML du W3C.

Une assertion entière peut être chiffrée dans cet élément et utilisée comme un identifiant. Dans ce cas, l'élément `<Subject>` de l'assertion chiffrée fournit l'"identifiant" du sujet de l'assertion d'enveloppe. Et donc, si l'assertion identifiante est non valide, l'assertion enveloppante l'est aussi.

Le fragment de schéma suivant définit l'élément `<EncryptedID>` et son type complexe **EncryptedElementType**:

```
<complexType name="EncryptedElementType">
    <sequence>
         <element ref="xenc:EncryptedData"/>
         <element ref="xenc:EncryptedKey" minOccurs="0"
maxOccurs="unbounded"/>
    </sequence>
</complexType>
<element name="EncryptedID" type="saml:EncryptedElementType"/>
```

8.1.2.5 Élément <Issuer>

L'élément `<Issuer>`, avec le type complexe **NameIDType**, donne des informations sur le producteur d'une assertion ou message de protocole SAML. L'élément exige l'utilisation d'une chaîne pour porter le nom du producteur, mais permet diverses pièces de données descriptives (voir au § 8.1.2.2).

Outrepassant la règle habituelle pour ce type d'élément, si aucune valeur `Format` n'est fournie avec cet élément, la valeur `urn:oasis:names:tc:SAML:2.0:nameid-format:entity` est en vigueur (voir au § 8.1.2.2).

Le fragment de schéma suivant définit l'élément `<Issuer>`:

```
<element name="Issuer" type="saml:NameIDType"/>
```

8.1.3 Assertions

Les paragraphes suivants définissent la construction SAML qui contient des informations d'assertion ou fournit le moyen de se référer à une assertion existante.

8.1.3.1 Élément <AssertionIDRef>

L'élément `<AssertionIDRef>` fait référence à une assertion SAML par son identifiant unique. L'autorité spécifique qui a produit l'assertion ou auprès de laquelle l'assertion peut être obtenue n'est pas spécifiée au titre de la référence. Voir au paragraphe 8.2.3 un élément de protocole qui utilise une telle référence pour demander l'assertion correspondante.

Le fragment de schéma suivant définit l'élément `<AssertionIDRef>`:

```
<element name="AssertionIDRef" type="NCName"/>
```

8.1.3.2 Élément <AssertionURIRef>

L'élément `<AssertionURIRef>` fait référence à une assertion SAML par une référence d'URI. La référence d'URI peut être utilisée pour récupérer l'assertion correspondante d'une manière spécifique de la référence d'URI. Voir au § 7.3 des informations sur la façon dont cet élément est utilisé dans une liaison de protocole pour accomplir cela.

Le fragment de schéma suivant définit l'élément `<AssertionURIRef>`:

```
<element name="AssertionURIRef" type="anyURI"/>
```

8.1.3.3 Élément <Assertion>

L'élément `<Assertion>` est du type complexe **AssertionType**. Ce type spécifie les informations de base qui sont communes à toutes les assertions, y compris les éléments et attributs suivants:

– `Version` [Exigé]

  Version de cette assertion. L'identifiant pour la version de SAML définie dans la présente Recommandation est "2.0". La question des versions de SAML est discutée au § 8.3.

– `ID` [Exigé]

  Identifiant pour cette assertion. Il est du type **xs:ID**, et doit suivre les exigences spécifiées au § 7.3 pour l'unicité de l'identifiant.

– `IssueInstant` [Exigé]

  Heure de production en UTC, comme décrit au § 7.3.

– `<Issuer>` [Exigé]

  Autorité SAML qui accrédite l'assertion. Le producteur devrait être non ambigu pour les consommateurs d'assertions prévus.

  La présente Recommandation ne définit pas de relations particulières entre l'entité représentée par cet élément et le signataire de l'assertion (s'il en est). Toute exigence de cette sorte, imposée par un consommateur d'assertion qui utilise l'assertion ou par des profils spécifiques, est spécifique de l'application.

– `<ds:Signature>` [Facultatif]

  Signature XML qui protège l'intégrité de l'assertion et authentifie son producteur, comme décrit ci-dessous et au § 8.4.

– `<Subject>` [Facultatif]

  Sujet de la ou des déclarations contenues dans l'assertion.

– `<Conditions>` [Facultatif]

  Conditions qui doivent être évaluées en attestant de la validité de l'assertion et/ou en l'utilisant. Voir au paragraphe 8.1.5 des informations supplémentaires sur la façon d'évaluer les conditions.

– `<Advice>` [Facultatif]

  Informations supplémentaires se rapportant à l'assertion, qui aident au traitement dans certaines situations mais qui peuvent être ignorées par les applications qui ne comprennent pas le conseil ou ne souhaitent pas en faire usage.

  Zéro, un ou plusieurs des éléments de déclaration suivants:

– `<Statement>`

  Déclaration d'un type défini dans un schéma d'extension. Un attribut **xsi:type** doit être utilisé pour indiquer le type réel de déclaration.

– `<AuthnStatement>`

  Déclaration d'authentification.

– `<AuthzDecisionStatement>`

  Déclaration de décision d'autorisation.

– `<AttributeStatement>`

  Déclaration d'attribut.

Une assertion sans déclaration doit contenir un élément `<Subject>`. Une telle assertion identifie un principal d'une manière qui peut être référencée ou confirmée en utilisant les méthodes de SAML, mais n'atteste d'aucune information supplémentaire associée à ce principal.

Autrement, `<Subject>`, s'il est présent, identifie le sujet de toutes les déclarations contenues dans l'assertion. Si `<Subject>` est omis, les déclarations de l'assertion s'appliquent alors à un sujet ou aux sujets identifiés d'une façon spécifique de l'application ou du profil. SAML lui même ne définit pas de telles déclarations, et une assertion sans sujet n'a pas de signification définie dans la présente Recommandation.

Selon les exigences de protocoles ou profils particuliers, le producteur d'assertions SAML peut souvent avoir besoin d'être authentifié, et la protection de l'intégrité peut souvent être demandée. Authentification et intégrité du message peuvent être fournies par des mécanismes apportés par une liaison de protocole utilisée durant la livraison d'une assertion (voir le § 10). L'assertion SAML peut être signée, ce qui apporte à la fois l'authentification du producteur et la protection de l'intégrité.

Si une telle signature est utilisée, l'élément `<ds:Signature>` doit alors être présent, et un consommateur d'assertions doit vérifier que la signature est valide (c'est-à-dire que l'assertion n'a pas été altérée) conformément à la Signature XML du W3C. Si elle est invalide, le consommateur d'asertions ne doit pas s'appuyer sur le contenu de l'assertion. Si elle est valide, le consommateur d'assertions devrait alors évaluer la signature pour déterminer l'identité et la pertinence du producteur et peut continuer à traiter l'assertion conformément à la présente Recommandation et à ce qui lui semble approprié (par exemple, évaluer les conditions, avis, suivre les règles spécifiques du profil, et ainsi de suite).

Signées ou non signées, l'inclusion de déclarations multiples au sein d'une seule assertion est sémantiquement équivalente à un ensemble d'assertions contenant individuellement des déclarations (pourvu que le sujet, les conditions, etc. soient aussi les mêmes).

Le fragment de schéma suivant définit l'élément `<Assertion>` et son type complexe **AssertionType**:

```
<element name="Assertion" type="saml:AssertionType"/>
<complexType name="AssertionType">
       <sequence>
              <element ref="saml:Issuer"/>
              <element ref="ds:Signature" minOccurs="0"/>
              <element ref="saml:Subject" minOccurs="0"/>
              <element ref="saml:Conditions" minOccurs="0"/>
              <element ref="saml:Advice" minOccurs="0"/>
              <choice minOccurs="0" maxOccurs="unbounded">
                     <element ref="saml:Statement"/>
                     <element ref="saml:AuthnStatement"/>
                     <element ref="saml:AuthzDecisionStatement"/>
                     <element ref="saml:AttributeStatement"/>
              </choice>
       </sequence>
       <attribute name="Version" type="string" use="required"/>
       <attribute name="ID" type="ID" use="required"/>
       <attribute name="IssueInstant" type="dateTime" use="required"/>
</complexType>
```

8.1.3.4 Élément <EncryptedAssertion>

L'élément <EncryptedAssertion> représente une assertion en mode chiffré, comme défini dans Chiffrement du W3C. L'élément <EncryptedAssertion> contient les éléments suivants:

– <xenc:EncryptedData> [Exigé]

  Contenu chiffré et détails de chiffrement associés, comme défini dans Chiffrement du W3C. L'attribut Type devrait être présent et, s'il est présent, doit contenir une valeur de http://www.w3.org/2001/04/xmlenc#Element. Le contenu chiffré doit contenir un élément qui a un type de **AssertionType** ou qui en est dérivé.

– <xenc:EncryptedKey> [Zéro, une ou plusieurs]

  Clés de déchiffrement enveloppées, comme défini dans Chiffrement du W3C. Chaque clé enveloppée devrait inclure un attribut Recipient qui spécifie l'entité pour laquelle la clé a été chiffrée. La valeur de l'attribut Recipient devrait être l'identifiant d'URI d'une entité système SAML comme défini par le § 8.7.

Les assertions chiffrées sont destinées à assurer un mécanisme de protection de la confidentialité lorsque la valeur du texte en clair passe à travers un intermédiaire.

Le fragment de schéma suivant définit l'élément <EncryptedAssertion>:

```
<element name="EncryptedAssertion" type="saml:EncryptedElementType"/>
```

8.1.4 Sujets

Le présent paragraphe définit la construction SAML utilisée pour décrire le sujet d'une assertion. L'élément facultatif <Subject> spécifie le principal qui est le sujet de toutes (zéro, une ou plusieurs) les déclarations contenues dans l'assertion. Il contient un identifiant, une série d'une ou plusieurs confirmations de sujet, ou les deux:

– <BaseID>, <NameID>, ou <EncryptedID> [Facultatif]

  Identifie le sujet.

– <SubjectConfirmation> [Zéro, une ou plusieurs]

  Informations qui permettent au sujet d'être confirmé. Si plus d'une confirmation de sujet est fournie, satisfaire l'une d'entre elles est alors suffisant pour confirmer le sujet pour les besoins de l'application de l'assertion.

Un élément <Subject> peut contenir à la fois un identifiant et zéro, une ou plusieurs confirmations de sujet qu'un consommateur d'assertions peut vérifier lors du traitement d'une assertion. Si une quelconque des confirmations de sujet incluses est vérifiée, le consommateur d'assertions peut traiter l'entité qui présente l'assertion comme une de celles que le producteur d'assertions a associées au principal identifié dans l'identifiant de nom et associées aux déclarations contenues dans l'assertion. Cette entité témoin et le sujet réel peuvent être ou non la même entité.

Si il n'y a pas de confirmation de sujet incluse, toute relation entre le présentateur de l'assertion et le sujet réel est non spécifiée.

Un élément <Subject> ne devrait pas identifier plus d'un principal.

Le fragment de schéma suivant définit l'élément <Subject> et son type complexe **SubjectType**:

```
<element name="Subject" type="saml:SubjectType"/>
<complexType name="SubjectType">
      <choice>
       <sequence>
              <choice>
                     <element ref="saml:BaseID"/>
                     <element ref="saml:NameID"/>
                     <element ref="saml:EncryptedID"/>
              </choice>
              <element ref="saml:SubjectConfirmation" minOccurs="0"
maxOccurs="unbounded"/>
       </sequence>
       <element ref="saml:SubjectConfirmation" maxOccurs="unbounded"/>
      </choice>
</complexType>
```

8.1.4.1 Élément <SubjectConfirmation>

L'élément <SubjectConfirmation> fournit à un consommateur d'assertions le moyen de vérifier la correspondance entre le sujet de l'assertion et la partie avec laquelle le consommateur d'assertions est en train de communiquer. Il contient les attributs et éléments suivants:

– Method [Exigé]

Référence d'URI qui identifie un protocole ou mécanisme à utiliser pour confirmer le sujet. Les références d'URI qui identifient les méthodes de confirmation définies par SAML sont définies au § 11. Des méthodes supplémentaires peuvent être ajoutées en définissant de nouveaux URI et profils ou par accord privé.

– <BaseID>, <NameID>, ou <EncryptedID> [Facultatif]

Identifie l'entité dont on attend qu'elle satisfasse les exigences de confirmation de sujet englobées.

– <SubjectConfirmationData> [Facultatif]

Informations supplémentaires de confirmation qui seront utilisées par une méthode de confirmation spécifique. Par exemple, le contenu normal de cet élément pourrait être un élément <ds:KeyInfo> comme défini dans Chiffrement du W3C, qui identifie une clé cryptographique (voir aussi au § 8.1.4.3). Des méthodes de confirmation particulières peuvent définir un type de schéma pour décrire les éléments, attributs, ou contenus qui peuvent apparaître dans l'élément <SubjectConfirmationData>.

Le fragment de schéma suivant définit l'élément <SubjectConfirmation> et son type complexe **SubjectConfirmationType**:

```
<element name="SubjectConfirmation" type="saml:SubjectConfirmationType"/>
<complexType name="SubjectConfirmationType">
      <sequence>
            <choice minOccurs="0">
                  <element ref="saml:BaseID"/>
                  <element ref="saml:NameID"/>
                  <element ref="saml:EncryptedID"/>
            </choice>
            <element ref="saml:SubjectConfirmationData" minOccurs="0"/>
      </sequence>
      <attribute name="Method" type="anyURI" use="required"/>
</complexType>
```

8.1.4.2 Élément <SubjectConfirmationData>

L'élément <SubjectConfirmationData> a le type complexe **SubjectConfirmationDataType**. Il spécifie des données supplémentaires qui permettent au sujet d'être confirmé, ou qui encadrent les circonstances dans lesquelles l'acte de confirmation de sujet peut avoir lieu. La confirmation de sujet a lieu lorsque un producteur d'assertions cherche à vérifier les relations entre l'entité qui a présenté l'assertion (c'est-à-dire, l'entité témoin) et le sujet de la revendication de l'assertion. Il contient les attributs facultatifs suivants qui peuvent s'appliquer à toute méthode:

– NotBefore [Facultatif]

Instant dans le temps avant lequel le sujet ne peut pas être confirmé. La valeur horaire est codée en UTC, comme décrit au § 7.3.

– NotOnOrAfter [Facultatif]

Instant auquel le sujet ne peut plus être confirmé. La valeur horaire est codée en UTC, comme décrit au § 7.3.

– Recipient [Facultatif]

URI qui spécifie l'entité ou la localisation à laquelle une entité témoin peut présenter l'assertion. Par exemple, cet attribut peut indiquer que l'assertion doit être livrée à un point de terminaison de réseau particulier afin d'empêcher un intermédiaire de le rediriger ailleurs.

– InResponseTo [Facultatif]

ID d'un message de protocole SAML en réponse auquel une entité témoin peut présenter l'assertion. Par exemple, cet attribut peut être utilisé pour corréler l'assertion pour une demande SAML qui a eu pour résultat sa présentation.

– Address [Facultatif]

Adresse/localisation réseau à partir de laquelle une entité témoin peut présenter l'assertion. Par exemple, cet attribut peut être utilisé pour lier l'assertion à des adresses de client particulières pour empêcher un attaquant de voler facilement l'assertion et de la présenter à partir d'une autre localisation. Les adresses IPv4 devraient

être représentées dans le format usuel en décimal séparé par des points (par exemple, "1.2.3.4"). Les adresses IPv6 devraient être représentées comme défini au § 2.2 de la RFC 3513 de l'IETF (par exemple, "FEDC:BA98:7654:3210:FEDC:BA98:7654:3210").

– `Attributs arbitraires`

  Ce type complexe utilise un point d'extension <xs:anyAttribute> pour permettre d'ajouter des attributs XML arbitraires qualifiés en espace de nom aux constructions <SubjectConfirmationData> sans qu'il soit besoin d'extension de schéma explicite. Cela permet d'ajouter autant de champs supplémentaires que nécessaire pour fournir des informations supplémentaires en rapport avec la confirmation. Les extensions SAML ne doivent pas ajouter d'attributs XML local (non qualifiés en espace de nom) ou d'attributs XML qualifiés par un espace de nom qualifié XML au type complexe **SubjectConfirmationDataType** ou à un type qui en est dérivé; de tels attributs sont réservés pour la maintenance et l'amélioration future de SAML lui-même.

– `Éléments arbitraires`

  Ce type complexe utilise un point d'extension <xs:any> pour permettre d'ajouter des éléments XML arbitraires aux constructions <SubjectConfirmationData> sans qu'il soit besoin d'une extension de schéma explicite. Ceci permet d'ajouter autant d'éléments supplémentaires que nécessaire pour fournir des informations supplémentaires en rapport avec la confirmation.

Des méthodes et profils de confirmation particuliers qui font usage de ces méthodes peuvent requérir l'usage d'un ou plusieurs des attributs définis dans ce type complexe. Par exemple, combien de ces attributs (et de confirmation de sujet en général) peuvent être utilisés, voir le § 13.

La durée spécifiée par les attributs facultatifs `NotBefore` et `NotOnOrAfter`, s'il sont présents, devrait tomber dans la période de validité globale de l'assertion, comme spécifié par les attributs `NotBefore` et `NotOnOrAfter` de l'élément <Conditions>. Si les deux attributs sont présents, la valeur de `NotBefore` doit être inférieure (plus tôt que) à la valeur de `NotOnOrAfter`.

Le fragment de schéma suivant définit l'élément <SubjectConfirmationData> et son type complexe **SubjectConfirmationDataType**:

```
<element name="SubjectConfirmationData"
type="saml:SubjectConfirmationDataType"/>
<complexType name="SubjectConfirmationDataType" mixed="true">
       <complexContent>
              <restriction base="anyType">
                     <sequence>
                            <any namespace="##any" processContents="lax"
minOccurs="0" maxOccurs="unbounded"/>
                     </sequence>
                     <attribute name="NotBefore" type="dateTime"
use="optional"/>
                     <attribute name="NotOnOrAfter" type="dateTime"
use="optional"/>
                     <attribute name="Recipient" type="anyURI"
use="optional"/>
                     <attribute name="InResponseTo" type="NCName"
use="optional"/>
                     <attribute name="Address" type="string"
use="optional"/>
                     <anyAttribute namespace="##other"
processContents="lax"/>
              </restriction>
       </complexContent>
</complexType>
```

8.1.4.3 Type complexe KeyInfoConfirmationDataType

Le type complexe **KeyInfoConfirmationDataType** oblige un élément <SubjectConfirmationData> à contenir un ou plusieurs éléments <ds:KeyInfo> qui identifient les clés cryptographiques qui sont utilisées d'une certaine façon pour authentifier une entité témoin. La méthode de confirmation particulière doit définir le mécanisme exact par lequel peuvent être utilisées les données de confirmation. Les attributs facultatifs définis par le type complexe **SubjectConfirmationDataType** peuvent aussi apparaître.

Ce type complexe, ou un type dérivé de lui, devrait être utilisé par toute méthode de confirmation qui définit ses données de confirmation en termes d'élément <ds:KeyInfo>.

Conformément à Chiffrement du W3C, chaque élément <ds:KeyInfo> doit identifier une seule clé cryptographique. Plusieurs clés peuvent être identifiées par des éléments <ds:KeyInfo> distincts, comme lorsque un principal utilise différentes clés pour se confirmer lui-même auprès de consommateurs d'assertions différents.

Le fragment de schéma suivant définit le type complexe **KeyInfoConfirmationDataType**:

```
<complexType name="KeyInfoConfirmationDataType" mixed="false">
	<complexContent>
		<restriction base="saml:SubjectConfirmationDataType">
			<sequence>
				<element ref="ds:KeyInfo" maxOccurs="unbounded"/>
			</sequence>
		</restriction>
	</complexContent>
</complexType>
```

8.1.4.4 Exemple d'un <Subject> confirmé par clé

Pour illustrer la façon dont divers éléments et types s'accordent ensemble figure ci-dessous un exemple d'un élément <Subject> contenant un identifiant de nom et une confirmation de sujet fondés sur la preuve de la possession d'une clé. Ici, l'utilisation de **KeyInfoConfirmationDataType** pour identifier la syntaxe de données de confirmation est un élément <ds:KeyInfo>:

```
<Subject>
	<NameID Format="urn:oasis:names:tc:SAML:1.1:nameid-
format:emailAddress">
	scott@example.org
	</NameID>
	<SubjectConfirmation Method="urn:oasis:names:tc:SAML:2.0:cm:holder-
of-key">
		<SubjectConfirmationData
xsi:type="saml:KeyInfoConfirmationDataType">
			<ds:KeyInfo>
				<ds:KeyName>Scott's Key</ds:KeyName>
			</ds:KeyInfo>
		</SubjectConfirmationData>
	</SubjectConfirmation>
</Subject>
```

8.1.5 Conditions

Le présent paragraphe définit la construction SAML qui met des contraintes à l'usage acceptable d'assertions SAML. L'élément <Conditions> peut contenir les éléments et attributs suivants:

– NotBefore [Facultatif]

  Spécifie le moment au plus tôt auquel l'assertion est valide. La valeur horaire est codée en UTC, comme décrit au paragraphe 7.3.

– NotOnOrAfter [Facultatif]

  Spécifie le moment auquel l'assertion est arrivée à expiration. La valeur horaire est codée en UTC, comme décrit au paragraphe 7.3.

– <Condition> [Tout nombre]

  Condition d'un type défini dans un schéma d'extension. Un attribut xsi:type doit être utilisé pour indiquer le type de condition réel.

– <AudienceRestriction> [Tout nombre]

  Spécifie que l'assertion est adressée à une audience particulière.

– <OneTimeUse> [Facultatif]

  Spécifie que l'assertion devrait être utilisée immédiatement et ne doit pas être conservée pour une utilisation future. Bien que le schéma permette plusieurs occurrences, il doit y avoir au plus une instance de cet élément.

– <ProxyRestriction> [Facultatif]

  Spécifie les limitations que le producteur d'assertions impose aux consommateurs d'assertions qui souhaitent agir ensuite eux-mêmes comme producteurs d'assertions et produire d'eux-même des assertions sur la base des

informations contenues dans l'assertion d'origine. Bien, que le schéma permette plusieurs occurrences, il doit y avoir au plus une instance de cet élément.

Parce que l'utilisation de l'attribut `xsi:type` permettrait à une assertion de contenir plus d'une instance d'un sous-type défini par SAML de **ConditionsType** (comme **OneTimeUseType**), le schéma ne limite pas explicitement le nombre de fois où des conditions particulières peuvent être incluses. Un type particulier de condition peut définir des limites à un tel usage, comme indiqué ci-dessus.

Le fragment de schéma suivant définit l'élément <Conditions> et son type complexe **ConditionsType**:

```
<element name="Conditions" type="saml:ConditionsType"/>
<complexType name="ConditionsType">
      <choice minOccurs="0" maxOccurs="unbounded">
            <element ref="saml:Condition"/>
            <element ref="saml:AudienceRestriction"/>
            <element ref="saml:OneTimeUse"/>
            <element ref="saml:ProxyRestriction"/>
      </choice>
      <attribute name="NotBefore" type="dateTime" use="optional"/>
      <attribute name="NotOnOrAfter" type="dateTime" use="optional"/>
</complexType>
```

8.1.5.1 Règles générales de traitement

Si une assertion contient un élément <Conditions>, la validité de l'assertion dépend alors des sous-éléments et attributs fournis, en utilisant les règles suivantes dans l'ordre indiqué ci-dessous.

Une assertion qui a un état de validité de condition "Valid" peut néanmoins être douteuse ou non valide pour des raisons telles que de n'être pas bien formée ou de schéma "valide", de n'être pas produite par une autorité SAML de confiance, ou de n'être pas authentifiée par un moyen de confiance.

Certaines conditions peuvent ne pas impacter directement la validité de l'assertion contenante (elles évaluent toujours comme "Valid"), mais peuvent contraindre le comportement des consommateurs d'assertions par rapport à l'utilisation de l'assertion:

- si aucun sous-élément ou attributs n'est fourni dans l'élément <Conditions>, l'assertion est alors considérée comme "Valid" par rapport au traitement de la condition;
- si un sous-élément ou attribut de l'élément <Conditions> est déterminé être non valide, l'assertion est alors considérée comme non valide;
- si un sous-élément ou attribut de l'élément <Conditions> ne peut pas être évalué, ou si un élément rencontré n'est pas compris, la validité de l'assertion ne peut être déterminée et est considérée comme "Indeterminate";
- si tous les sous-éléments et attributs de l'élément <Conditions> sont déterminés comme étant "Valid", l'assertion est alors considérée comme "Valid" par rapport au traitement de la condition.

La première règle qui s'applique termine le traitement des conditions; et donc la détermination qu'une assertion est "Invalid" prend le pas sur celle de "Indeterminate".

Une assertion déterminée comme "Invalid" ou "Indeterminate" doit être rejetée par un consommateur d'assertions (quel que soit le contexte ou profil de traitement), comme si l'assertion était mal formée ou inutilisable.

8.1.5.2 Attributs NotBefore et NotOnOrAfter

Les attributs `NotBefore` et `NotOnOrAfter` spécifient les limites de durée de validité de l'assertion dans le contexte de son ou ses profils d'utilisation. Ils ne garantissent pas que les déclarations contenues dans l'assertion seront correctes ou appropriées tout au long de la période de validité.

L'attribut `NotBefore` spécifie le moment auquel l'intervalle de validité commence. L'attribut `NotOnOrAfter` spécifie le moment où l'intervalle de validité se termine.

Si la valeur de `NotBefore` ou de `NotOnOrAfter` est omise, elle est alors considérée comme non spécifiée. Si l'attribut `NotBefore` est non spécifié (et si toutes les autres conditions fournies sont évaluées comme "Valid"), l'assertion est alors "Valid" par rapport aux conditions à tout moment antérieur à l'instant spécifié par l'attribut `NotOnOrAfter`. Si l'attribut `NotOnOrAfter` est non spécifié (et si toutes les autres conditions fournies sont évaluées comme "Valid"), l'assertion est "Valid" par rapport aux conditions à partir de l'instant spécifié par l'attribut `NotBefore` sans délai d'expiration. Si aucun des deux atrributs n'est spécifié (et toutes autres conditions fournies sont évaluées comme "Valid"), l'assertion est "Valid" par rapport aux conditions à tout moment.

Si les deux attributs sont présents, la valeur pour `NotBefore` doit être inférieure (plus tôt que) à la valeur pour `NotOnOrAfter`.

8.1.5.3 Élément <Condition>

L'élément `<Condition>` sert de point d'extension pour de nouvelles conditions. Son type complexe **ConditionAbstractType** est abstrait et n'est donc utilisable que comme base d'un type dérivé.

Le fragment de schéma suivant définit l'élément `<Condition>` et son type complexe **ConditionAbstractType**:

```
<element name="Condition" type="saml:ConditionAbstractType"/>
<complexType name="ConditionAbstractType" abstract="true"/>
```

8.1.5.4 Éléments <AudienceRestriction> et <Audience>

L'élément `<AudienceRestriction>` spécifie que l'assertion est adressée à une ou plusieurs audiences spécifiques identifiées par des éléments `<Audience>`. Bien qu'un consommateur d'assertions SAML se trouvant en dehors des audiences spécifiées soit capable de tirer des conclusions d'une assertion, le producteur d'assertions SAML ne fait aucune représentation explicite sur la pertinence ou sur la fiablilité à de tels consommateurs. Il contient l'élément suivant:

– `<Audience>`

  Référence d'URI qui identifie une audience de destination. La référence d'URI peut identifier un document qui décrit les termes et conditions de participation à l'audience. Il peut aussi contenir l'URI d'identifiant unique provenant d'un identifiant de nom SAML que décrit une entité système.

La condition de restriction d'audience évalue comme "Valid" si et seulement si le consommateur d'assertions SAML est un membre d'une ou plusieurs des audiences spécifiées.

Le producteur d'assertions SAML ne peut pas empêcher un consommateur à qui l'assertion est révélée d'entreprendre une action sur la base de l'information fournie. Cependant, l'élément `<AudienceRestriction>` permet au producteur d'assertions SAML de déclarer explicitement qu'aucune garantie n'est fournie à un tel consommateur dans une forme lisible par l'homme et par la machine. Alors qu'il n'y a aucune assurance qu'une cour de justice retienne une telle exclusion de garantie en toutes circonstances, la probabilité de maintien de l'exclusion de garantie est considérablement améliorée.

Plusieurs éléments <AudienceRestriction> peuvent être inclus dans une seule assertion, et chacune doit être évaluée de façon indépendante. Les effets de cette exigence et de la définition précédente sont soumis à une condition donnée, que les audiences forment des ensembles disjoints (un "OU" logique) alors que plusieurs conditions forment une conjonction (un "ET" logique).

Le fragment de schéma suivant définit l'élément <AudienceRestriction> et son type complexe **AudienceRestrictionType**:

```
<element name="AudienceRestriction"
       type="saml:AudienceRestrictionType"/>
<complexType name="AudienceRestrictionType">
       <complexContent>
              <extension base="saml:ConditionAbstractType">
                     <sequence>
                            <element ref="saml:Audience"
maxOccurs="unbounded"/>
                     </sequence>
              </extension>
       </complexContent>
</complexType>
<element name="Audience" type="anyURI"/>
```

8.1.5.5 Élément <OneTimeUse>

En général, les consommateurs d'assertions gardent les assertions, ou les informations qu'elles contiennent, sous une autre forme, pour les réutiliser. L'élément `<OneTimeUse>` permet à une autorité d'indiquer que les informations contenues dans l'assertion vont vraisemblablement changer très rapidement et que des information fraîches devraient être obtenues pour chaque utilisation. Ce serait par exemple une assertion contenant un `<AuthzDecisionStatement>` qui serait le résultat d'une politique spécifiant un contrôle d'accès en fonction de l'heure.

Si dans un environnement distribué les horloges système devaient être synchronisées avec précision, cette exigence pourrait être satisfaite en utilisant avec soin l'intervalle de validité. Cependant, comme il y aura toujours un certain biais d'horloge entre les systèmes et qu'il se combine à de possibles délais de transmission, il n'y a pas de méthode

convenable pour que le producteur limite la durée de vie d'une assertion sans courir un risque substantiel qu'elle soit déjà arrivée à expiration avant sa réception.

L'élément <OneTimeUse> indique que l'assertion devrait être utilisée immédiatement par le consommateur d'assertions et ne devrait pas être conservée pour utilisation ultérieure. Les consommateurs d'assertions ont toujours la liberté de demander une assertion fraîche pour chaque utilisation. Cependant, les implémentations qui choisissent de conserver les assertions pour utilisation ultérieure doivent respecter l'élément <OneTimeUse>. Cette condition est indépendante des informations de condition NotBefore et NotOnOrAfter.

Pour prendre en charge la contrainte d'usage unique, un consommateur d'assertions devrait maintenir une mémoire cache des assertions qu'il a traitées et qui contiennent une telle condition. Chaque fois qu'est traitée une assertion contenant cette condition, la mémoire cache devrait être sollicitée pour s'assurer que la même assertion n'a pas été reçue et traitée précédemment par le consommateur d'assertions.

Une autorité SAML ne doit pas inclure plus d'un élément <OneTimeUse> dans un élément <Conditions> d'une assertion.

Pour les besoins de la détermination de validité de l'élément <Conditions>, <OneTimeUse> est considéré comme étant toujours valide. C'est-à-dire que cette condition n'affecte pas la validité mais est une condition d'utilisation.

Le fragment de schéma suivant définit l'élément <OneTimeUse> et son type complexe **OneTimeUseType**:

```
<element name="OneTimeUse" type="saml:OneTimeUseType"/>
<complexType name="OneTimeUseType">
       <complexContent>
              <extension base="saml:ConditionAbstractType"/>
</complexContent>
</complexType>
```

8.1.5.6 Élément <ProxyRestriction>

Spécifie les limitations que le producteur d'assertions impose aux consommateurs d'assertions qui souhaitent à leur tour agir comme producteurs d'assertions et produire ensuite des assertions de leur propre chef sur la base des informations contenues dans l'assertion d'origine. Un consommateur d'assertions agissant comme producteur d'assertions ne doit pas produire d'assertions qui elles-mêmes violent les restrictions spécifiées dans cette condition sur la base d'une assertion contenant une telle condition.

L'élément <ProxyRestriction> contient les éléments et attributs suivants:

– Count [Facultatif]

  Spécifie le nombre maximum d'intermédiaires dont le producteur d'assertions permet l'existence entre cette assertion et une assertion qui en fin de compte a été produite en s'appuyant sur elle.

– <Audience> [Zéro, un, ou plusieurs]

  Spécifie l'ensemble des audiences à qui le producteur d'assertions permet de produire de nouvelles assertions sur la base de cette assertion.

Une valeur de Count compte de zéro indique qu'un consommateur d'assertions ne doit pas produire une assertion auprès d'un autre consommateur d'assertions sur la base de cette assertion. Si elle est supérieure à zéro, toute assertion produite de cette façon doit elle-même contenir un élément <ProxyRestriction> avec une valeur de Count compte au plus inférieure de un à cette valeur.

Si aucun élément <Audience> n'est spécifié, aucune restriction d'audience n'est imposée sur les consommateurs d'assertions à qui les assertions ultérieures peuvent être fournies. Autrement, toute assertion ainsi produite doit elle-même contenir un élément <AudienceRestriction> qui soit au moins un des éléments <Audience> présents dans l'élément <ProxyRestriction> précédent, et aucun élément <Audience> présent qui n'était pas dans l'élément <ProxyRestriction> précédent.

Une autorité SAML ne doit pas inclure plus d'un élément <ProxyRestriction> dans un élément <Conditions> d'une assertion.

Pour les besoins de la détermination de la validité de l'élément <Conditions>, la condition <ProxyRestriction> est considérée comme toujours valide. C'est-à-dire que cette condition n'affecte pas la validité mais est une condition d'utilisation.

Le fragment de schéma suivant définit l'élément <ProxyRestriction> et son type complexe **ProxyRestrictionType**:

```
<element name="ProxyRestriction" type="saml:ProxyRestrictionType"/>
<complexType name="ProxyRestrictionType">
       <complexContent>
              <extension base="saml:ConditionAbstractType">
                     <sequence>
                            <element ref="saml:Audience" minOccurs="0"
maxOccurs="unbounded"/>
                     </sequence>
                     <attribute name="Count" type="nonNegativeInteger"
use="optional"/>
              </extension>
       </complexContent>
</complexType>
```

8.1.6 Advice

Le présent paragraphe définit les constructions SAML qui contiennent des informations supplémentaires sur une assertion qu'un producteur d'assertions souhaite fournir à un consommateur d'assertions.

L'élément <Advice> contient toutes les informations supplémentaires que l'autorité SAML souhaite fournir. Ces informations peuvent être ignorées par les applications sans affecter la sémantique ou la validité de l'assertion.

L'élément <Advice> contient un mélange de zéro, un ou plusieurs éléments <Assertion>, <EncryptedAssertion>, <AssertionIDRef>, et <AssertionURIRef>, et d'éléments qualifiés par un espace de nom dans d'autres espaces de nom non-SAML.

Ci-après figurent des utilisations potentielles de l'élément <Advice>:

- inclure des preuves évidentes à l'appui des revendications de l'assertion citée, soit directement (en incorporant les revendications) soit indirectement (par référence aux assertions citées à l'appui);
- établir une preuve des revendications de l'assertion;
- spécifier l'horaire et les points de distribution pour les mises à jour de l'assertion.

Le fragment de schéma suivant définit l'élément <Advice> et son type complexe **AdviceType**:

```
<element name="Advice" type="saml:AdviceType"/>
<complexType name="AdviceType">
       <choice minOccurs="0" maxOccurs="unbounded">
              <element ref="saml:AssertionIDRef"/>
              <element ref="saml:AssertionURIRef"/>
              <element ref="saml:Assertion"/>
              <element ref="saml:EncryptedAssertion"/>
              <any namespace="##other" processContents="lax"/>
       </choice>
</complexType>
```

8.1.7 Déclarations

Toutes les déclarations définies par SAML sont associées à un sujet. Les assertions SAML sont habituellement faites à propos d'un **sujet**, représenté par l'élément <Subject>. Cependant, l'élément <Subject> est facultatif, et d'autres spécifications et profils peuvent utiliser la structure des assertions SAML pour faire des déclarations similaires sans spécifier un sujet, ou éventuellement en spécifiant le sujet d'une autre façon. Les paragraphes suivants définissent les constructions de SAML qui contiennent des informations de déclaration.

8.1.7.1 Élément <Statement>

L'élément <Statement> est un point d'extension qui permet à d'autres applications fondées sur l'assertion de réutiliser le cadre d'assertions SAML. SAML lui-même tire ses déclarations centrales de ce point d'extension. Son type complexe **StatementAbstractType** est abstrait et n'est donc utilisable que comme base d'un type dérivé.

Le fragment de schéma suivant définit l'élément <Statement> et son type complexe **StatementAbstractType**:

```
<element name="Statement" type="saml:StatementAbstractType"/>
<complexType name="StatementAbstractType" abstract="true"/>
```

8.1.7.2 Élément <AuthnStatement>

L'élément <AuthnStatement> décrit une déclaration d'une autorité SAML certifiant que le sujet de l'assertion a été authentifié par un moyen particulier à un moment donné. Les assertions contenant des éléments <AuthnStatement> doivent contenir un élément <Subject>.

Il est du type **AuthnStatementType**, qui étend **StatementAbstractType** avec l'ajout des éléments et attributs suivants:

NOTE – L'élément <AuthorityBinding> et son type correspondant ont été retirés de <AuthnStatement> pour SAML V2.0.

– AuthnInstant [Exigé]

  Spécifie le moment auquel l'authentification a eu lieu. La valeur horaire est codée en UTC, comme décrit au paragraphe 7.3.

– SessionIndex [Facultatif]

  Spécifie l'indice d'une session donnée entre le principal identifié par le sujet et l'autorité d'authentification.

– SessionNotOnOrAfter [Facultatif]

  Spécifie un instant auquel la session entre le principal identifié par le sujet et l'autorité SAML produisant cette déclaration doit être considérée comme terminée. La valeur horaire est codée en UTC, comme décrit au paragraphe 7.3. Il n'y a pas de relation obligatoire entre cet attribut et un attribut de condition NotOnOrAfter qui peut être présent dans l'assertion.

– <SubjectLocality> [Facultatif]

  Spécifie le nom de domaine DNS et l'adresse IP pour le système à partir duquel le sujet d'assertion a été apparemment authentifié.

– <AuthnContext> [Exigé]

  Le contexte utilisé par l'autorité d'authentification jusqu'à et y compris l'événement d'authentification qui donne cette déclaration. Il contient une référence de classe de contexte d'authentification, une déclaration de contexte d'authentification ou une référence de déclaration, ou les deux. Voir au § 12 (Contexte d'authentification) une description complète des informations de contexte d'authentification.

En général, toute valeur de chaîne peut être utilisée comme valeur de SessionIndex. Cependant, lorsqu'on prend en considération la confidentialité, il faut veiller à s'assurer que la valeur de SessionIndex ne rend pas non valides d'autres mécanismes de confidentialité. En conséquence, la valeur ne devrait pas être utilisable pour corréler l'activité par un principal à travers différents participants de session. Deux solutions sont fournies ci-dessous pour réaliser cet objectif, et elles sont recommandées:

- utiliser de petits entiers positifs (ou rendre des constantes récurrentes dans une liste) pour SessionIndex. L'autorité SAML devrait choisir la gamme des valeurs de telle sorte que la cardinalité de tout entier soit suffisamment élevée pour empêcher les actions d'un principal particulier d'être corrélées à travers plusieurs participants d'une session. L'autorité SAML devrait choisir de façon aléatoire des valeurs pour SessionIndex au sein de cette gamme (excepté quand il faut assurer des valeurs univoques pour les déclarations ultérieures données au même participant de session mais au titre d'une session distincte);
- utiliser la valeur d'identifiant d'assertion enveloppante dans le SessionIndex.

Le fragment de schéma suivant définit l'élément <AuthnStatement> et son type complexe **AuthnStatementType**:

```
<element name="AuthnStatement" type="saml:AuthnStatementType"/>
<complexType name="AuthnStatementType">
       <complexContent>
              <extension base="saml:StatementAbstractType">
                     <sequence>
                            <element ref="saml:SubjectLocality"
minOccurs="0"/>
                            <element ref="saml:AuthnContext"/>
                     </sequence>
                     <attribute name="AuthnInstant" type="dateTime"
use="required"/>
                     <attribute name="SessionIndex" type="string"
use="optional"/>
                     <attribute name="SessionNotOnOrAfter" type="dateTime"
use="optional"/>
              </extension>
       </complexContent>
</complexType>
```

8.1.7.2.1 Élément <SubjectLocality>

L'élément <SubjectLocality> spécifie le nom de domaine DNS et l'adresse IP pour le système à partir duquel le sujet d'assertion a été authentifié. Il a les attributs suivants:

– Address [Facultatif]

  Adresse réseau du système à partir duquel le principal identifié par le sujet a été authentifié. Les adresses IPv4 devraient être représentées en format décimal séparé par des points (par exemple, "1.2.3.4"). Les adresses IPv6 devraient être représentées comme défini au § 2.2 de la RFC 3513 de l'IETF (par exemple, "FEDC:BA98:7654:3210:FEDC:BA98:7654:3210").

– DNSName [Facultatif]

  Nom DNS du système à partir duquel le principal identifié par le sujet a été authentifié.

Cet élément est entièrement informatif, car ces deux champs sont très facilement "parodiés", mais ce peuvent être des informations utiles dans certaines applications.

Le fragment de schéma suivant définit l'élément <SubjectLocality> et son type complexe **SubjectLocalityType**:

```
<element name="SubjectLocality" type="saml:SubjectLocalityType"/>
<complexType name="SubjectLocalityType">
      <attribute name="Address" type="string" use="optional"/>
      <attribute name="DNSName" type="string" use="optional"/>
</complexType>
```

8.1.7.2.2 Élément <AuthnContext>

L'élément <AuthnContext> spécifie le contexte d'un événement d'authentification. L'élément peut contenir une référence de classe de contexte d'authentification, une déclaration de contexte d'authentification ou référence de déclaration, ou les deux. Son type complexe **AuthnContextType** a les éléments suivants:

– <AuthnContextClassRef> [Facultatif]

  Référence d'URI qui identifie une classe de contexte d'authentification qui décrit la déclaration de contexte d'authentification qui suit.

– <AuthnContextDecl> ou <AuthnContextDeclRef> [Facultatif]

  Soit une déclaration de contexte d'authentification fournie par une valeur, soit une référence d'URI qui identifie une telle déclaration. La référence d'URI peut se résoudre directement en un document XML contenant la déclaration référencée.

– <AuthenticatingAuthority> [Zéro, un ou plusieurs]

  Zéro, un ou plusieurs identifiants uniques d'autorités d'authentification qui ont été impliquées dans l'authentification du principal (non inclus le producteur de l'assertion, qui est présumé avoir été impliqué sans qu'il soit explicitement nommé ici).

Voir au paragraphe 12 une description complète des informations de contexte d'authentification.

Le fragment de schéma suivant définit l'élément <AuthnContext> et son type complexe **AuthnContextType**:

```
<element name="AuthnContext" type="saml:AuthnContextType"/>
<complexType name="AuthnContextType">
      <sequence>
            <choice>
                  <sequence>
                        <element ref="saml:AuthnContextClassRef"/>
                        <choice minOccurs="0">
                              <element ref="saml:AuthnContextDecl"/>
                              <element ref="saml:AuthnContextDeclRef"/>
                        </choice>
                  </sequence>
                  <choice>
                        <element ref="saml:AuthnContextDecl"/>
                        <element ref="saml:AuthnContextDeclRef"/>
                  </choice>
            </choice>
            <element ref="saml:AuthenticatingAuthority" minOccurs="0"
maxOccurs="unbounded"/>
      </sequence>
</complexType>
```

```
<element name="AuthnContextClassRef" type="anyURI"/>
<element name="AuthnContextDeclRef" type="anyURI"/>
<element name="AuthnContextDecl" type="anyType"/>
<element name="AuthenticatingAuthority" type="anyURI"/>
```

8.1.7.3 Élément <AttributeStatement>

L'élément <AttributeStatement> décrit une déclaration d'une autorité SAML certifiant que le sujet d'assertion est associé aux attributs spécifiés. Les assertions qui contiennent des éléments <AttributeStatement> doivent contenir un élément <Subject>.

Il est du type **AttributeStatementType**, qui étend **StatementAbstractType** avec l'ajout des éléments suivants:

– <Attribute> ou <EncryptedAttribute> [Un ou plusieurs]

L'élément <Attribute> spécifie un attribut du sujet d'assertion. Un attribut SAML chiffré peut être inclus avec l'élément <EncryptedAttribute>.

Le fragment de schéma suivant définit l'élément <AttributeStatement> et son type complexe **AttributeStatementType**:

```
<element name="AttributeStatement" type="saml:AttributeStatementType"/>
<complexType name="AttributeStatementType">
      <complexContent>
            <extension base="saml:StatementAbstractType">
                  <choice maxOccurs="unbounded">
                        <element ref="saml:Attribute"/>
                        <element ref="saml:EncryptedAttribute"/>
                  </choice>
            </extension>
      </complexContent>
</complexType>
```

8.1.7.3.1 Élément <Attribute>

L'élément <Attribute> identifie un attribut par son nom et inclut facultativement sa ou ses valeurs. Il a le type complexe **AttributeType**. Il est utilisé au sein d'une déclaration d'attribut pour exprimer des attributs et valeurs particuliers associés à un sujet d'assertion, comme décrit au paragraphe précédent. Il sert aussi dans une interrogation d'attribut pour demander que les valeurs d'attributs SAML spécifiques soient retournées. L'élément <Attribute> contient les attributs XML suivants:

– Name [Exigé]

Nom de l'attribut.

– NameFormat [Facultatif]

Référence d'URI représentant la classification du nom de l'attribut pour les besoins de l'interprétation du nom. Voir au § 8.7.2 quelques références d'URI qui peuvent être utilisées comme valeur de l'attribut NameFormat et leur description et règles de traitement associées. Si aucune valeur de NameFormat n'est fournie, c'est l'identifiant urn:oasis:names:tc:SAML:2.0:attrname-format:unspecified qui est en vigueur.

– FriendlyName [Facultatif]

Chaîne qui fournit une forme lisible du nom de l'attribut, qui peut être utile dans les cas où le Name nom réel est complexe ou opaque, comme un OID (comme défini dans la Rec. UIT-T X.660) ou un UUID (comme défini dans la Rec. UIT-T X.667). Cette valeur d'attribut ne doit pas être utilisée comme base d'identification formelle des attributs SAML.

– Attributs arbitraires

Ce type complexe utilise un point d'extension <xs : anyAttribute> pour permettre l'ajout d'attributs XML arbitraires aux constructions <Attribute> sans avoir besoin d'une extension de schéma explicite. Cela permet d'ajouter en tant que de besoin des champs supplémentaires pour fournir des paramètres additionnels, à utiliser, par exemple, dans une interrogation d'attributs. Les extensions SAML ne doivent pas ajouter d'attributs XML locaux (non qualifiés d'espace de nom) ou d'attributs XML qualifiés par un espace de nom défini par SAML au type complexe **AttributeType** ou à un type qui en est dérivé; de tels attributs sont réservés pour la maintenance et l'amélioration futures de SAML lui-même.

– <AttributeValue> [Nombre quelconque]

Contient une valeur de l'attribut. Si un attribut contient plus d'une valeur discrète, il est recommandé que chaque valeur apparaisse sous sa forme d'élément <AttributeValue>. Si plus d'un élément <AttributeValue> est fourni pour un attribut, et si un des éléments a un datatype *(type de données)* alloué au moyen de xsi:type, alors tous les éléments <AttributeValue> doivent avoir alloué un datatype identique.

La signification d'un élément <Attribute> qui ne contient aucun élément <AttributeValue> dépend de son contexte. Au sein d'un <AttributeStatement>, si l'attribut SAML existe mais n'a pas de valeur, l'élément <AttributeValue> doit être omis. Au sein de <samlp:AttributeQuery>, l'absence de valeur indique que le demandeur est intéressé par une ou toutes les valeurs nommées de l'attribut (voir aussi au § 8.2).

Toute autre utilisation de l'élément <Attribute> par des profils ou autres spécifications doit définir la sémantique de la spécification ou de l'omission des éléments <AttributeValue>.

Le fragment de schéma suivant définit l'élément <Attribute> et son type complexe **AttributeType**:

```
<element name="Attribute" type="saml:AttributeType"/>
<complexType name="AttributeType">
       <sequence>
              <element ref="saml:AttributeValue" minOccurs="0"
maxOccurs="unbounded"/>
       </sequence>
       <attribute name="Name" type="string" use="required"/>
       <attribute name="NameFormat" type="anyURI" use="optional"/>
       <attribute name="FriendlyName" type="string" use="optional"/>
       <anyAttribute namespace="##other" processContents="lax"/>
</complexType>
```

L'élément <AttributeValue> fournit la valeur d'un attribut SAML spécifié. Il est du type **xs:anyType**, qui permet à tout XML bien formé d'apparaître comme contenu de l'élément.

Si le contenu des données d'un élément <AttributeValue> est d'un type simple de schéma XML (tel que **xs:integer** ou **xs:string**), le datatype peut être déclaré explicitement au moyen d'une déclaration xsi:type dans l'élément <AttributeValue>. Si la valeur d'attribut contient des données structurées, les éléments de données nécessaires peuvent être définis dans un schéma d'extension.

NOTE – Spécifier un datatype autre qu'un type simple de schéma XML sur <AttributeValue> en utilisant xsi:type exigera la présence du schéma d'extension qui définit le datatype afin de permettre la poursuite du traitement de schéma.

Si un attribut SAML inclut une valeur vide, telle qu'une chaîne vide, l'élément <AttributeValue> correspondant doit être vide (cela se présente généralement sous la forme <AttributeValue/>). Cela prend le pas sur l'exigence du § 7.1 que les valeurs de chaînes dans un contenu SAML contiennent au moins un caractère non espace blanc.

Si un attribut SAML inclut une valeur "null", l'élément <AttributeValue> correspondant doit être vide et doit contenir l'attribut XML réservé xsi:nil avec une valeur de "true" ou "1".

Le fragment de schéma suivant définit l'élément <AttributeValue>:

```
<element name="AttributeValue" type="anyType" nillable="true"/>
```

8.1.7.3.2 Élément <EncryptedAttribute>

L'élément <EncryptedAttribute> représente un attribut SAML en mode chiffré, comme défini dans Chiffrement du W3C. L'élément <EncryptedAttribute> contient les éléments suivants:

– <xenc:EncryptedData> [Exigé]

Contenu chiffré et détails de chiffrement associés, comme défini dans Chiffrement du W3C. L'attribut Type devrait être présent et, s'il est présent, doit contenir une valeur de http://www.w3.org/2001/04/xmlenc#Element. Le contenu chiffré doit contenir un élément ayant un type de **AttributeType** ou qui en soit dérivé.

– <xenc:EncryptedKey> [Zéro, une ou plusieurs]

Clés de déchiffrement enveloppées, comme défini dans Chiffrement du W3C. Chaque clé enveloppée devrait inclure un attribut Recipient qui spécifie l'entité pour laquelle la clé a été chiffrée. La valeur de l'attribut Recipient devrait être l'identifiant d'URI d'une entité système avec un identifiant de nom SAML, comme défini au § 8.7.

Les attributs chiffrés sont destinés à assurer une protection de la confidentialité lorsque la valeur du texte en clair passe par un intermédiaire.

Le fragment de schéma suivant définit l'élément <EncryptedAttribute>:

```
<element name="EncryptedAttribute" type="saml:EncryptedElementType"/>
```

8.1.7.4 Élément <AuthzDecisionStatement>

L'élément <AuthzDecisionStatement> décrit une déclaration d'une autorité SAML certifiant qu'une demande d'accès par le sujet d'assertion à la ressource spécifiée a eu pour résultat la décision d'autorisation spécifiée sur la base d'une évidence dont la spécification est facultative. Les assertions contenant des éléments <AuthzDecisionStatement> doivent contenir un élément <Subject>.

La ressource est identifiée au moyen d'une référence d'URI. Pour que l'assertion soit interprétée correctement et en toute sécurité, l'autorité SAML et le consommateur d'assertions SAML doivent interpréter chaque référence d'URI de façon cohérente. L'échec de l'interprétation cohérente d'une référence d'URI peut avoir pour résultat des décisions d'autorisation différentes selon le codage de la référence d'URI de la ressource. Les règles de normalisation des références d'URI figurent au § 6 de la RFC 2396 de l'IETF.

Pour éviter les ambiguïtés résultant des variations du codage d'URI, les entités système SAML devraient employer chaque fois que possible la forme normalisée d'URI de la façon suivante:

- les autorités SAML devraient coder toutes les références d'URI de ressources en forme normalisée;
- les consommateurs d'assertions devraient convertir les références d'URI de ressources en forme normalisée avant de commencer le traitement.

L'interprétation de références d'URI non cohérentes peut aussi résulter de différences entre la syntaxe de référence d'URI et la sémantique d'un système de fichiers sous-jacent. Un soin particulier est nécessaire si les références d'URI servent à spécifier un langage de politique de contrôle d'accès. Les conditions de sécurité suivantes devraient être satisfaites par le système qui utilise les assertions SAML:

- des parties de la syntaxe de la référence d'URI sont sensibles à la casse. Si le système de fichiers sous-jacent n'est pas sensible à la casse, un demandeur ne devrait pas être capable d'obtenir l'accès à une ressource refusée en changeant la casse d'une partie de la référence d'URI de la ressource;
- de nombreux systèmes de fichiers prennent en charge des mécanismes tels que des chemins logiques et des liens symboliques, ce qui permet aux utilisateurs d'établir des équivalences logiques entre les entrées de système de fichier. Un demandeur ne devrait pas être capable d'obtenir l'accès à une ressource refusée en créant une telle équivalence.

L'élément <AuthzDecisionStatement> est du type **AuthzDecisionStatementType**, qui étend **StatementAbstractType** avec l'ajout des éléments et attributs suivants:

– Resource [Exigé]

  Référence d'URI qui identifie la ressource à laquelle l'autorisation d'accès est demandée. Cet attribut peut avoir la valeur d'une référence d'URI vide (""), et sa signification est définie comme étant "le début du document en cours", comme spécifié au § 4.2 de la RFC 2396 de l'IETF.

– Decision [Exigé]

  Décision rendue par l'autorité SAML par rapport à la ressource spécifiée. La valeur est du type simple **DecisionType**.

– <Action> [Une ou plusieurs]

  Ensemble des actions autorisées sur la ressource spécifiée.

– <Evidence> [Facultatif]

  Ensemble des assertions sur lesquelles l'autorité SAML s'appuie pour prendre la décision.

Le fragment de schéma suivant définit l'élément <AuthzDecisionStatement> et son type complexe **AuthzDecisionStatementType**:

```
<element name="AuthzDecisionStatement"
 type="saml:AuthzDecisionStatementType"/>
<complexType name="AuthzDecisionStatementType">
 <complexContent>
  <extension base="saml:StatementAbstractType">
         <sequence>
```

```
                <element ref="saml:Action" maxOccurs="unbounded"/>
                <element ref="saml:Evidence" minOccurs="0"/>
        </sequence>
        <attribute name="Resource" type="anyURI" use="required"/>
        <attribute name="Decision" type="saml:DecisionType" use="required"/>
  </extension>
 </complexContent>
</complexType>
```

8.1.7.4.1 Type simple de DecisionType

Le type simple de **DecisionType** définit les valeurs possibles à rapporter comme état d'une déclaration de décision d'autorisation.

– `Permit`

  L'action spécifiée est permise.

– `Deny`

  L'action spécifiée est refusée.

– `Indeterminate`

  L'autorité SAML ne peut pas déterminer si l'action spécifiée est permise ou refusée.

La valeur de décision `Indeterminate` est utilisée dans des situations où l'autorité SAML exige la capacité de fournir une déclaration affirmative mais n'est pas capable de produire une décision. Des informations supplémentaires sur les raisons du refus ou de l'incapacité à fournir une décision peuvent être retournées comme éléments `<StatusDetail>` dans la `<Response>` qui l'enveloppe.

Le fragment de schéma suivant définit le type simple **DecisionType**:

```
<simpleType name="DecisionType">
       <restriction base="string">
              <enumeration value="Permit"/>
              <enumeration value="Deny"/>
              <enumeration value="Indeterminate"/>
       </restriction>
</simpleType>
```

8.1.7.4.2 Élément <Action>

L'élément `<Action>` spécifie une action sur la ressource spécifiée pour laquelle la permission est demandée. Son contenu de chaîne de données fournit le nom de l'action recherchée à effectuer sur la ressource spécifiée, et il a l'attribut suivant:

– `Namespace` [Facultatif]

  Référence d'URI représentant l'espace de nom dans lequel est à interpréter l'action spécifiée. Si cet élément est absent, l'espace de nom `urn:oasis:names:tc:SAML:1.0:action:rwedc-negation` spécifié au § 8.7 est en vigueur.

  NOTE (informative) – PE 36 (voir OASIS PE:2006) suggère de remplacer le texte ci-dessus par:

  Namespace [Exigé]

  Référence d'URI représentant l'espace de nom dans lequel le nom de l'action spécifiée est à interpréter.

Le fragment de schéma suivant définit l'élément `<Action>` et son type complexe **ActionType**:

```
<element name="Action" type="saml:ActionType"/>
<complexType name="ActionType">
 <simpleContent>
       <extension base="string">
              <attribute name="Namespace" type="anyURI" use="required"/>
       </extension>
 </simpleContent>
</complexType>
```

8.1.7.4.3 Élément <Evidence>

L'élément `<Evidence>` contient une ou plusieurs assertions ou références d'assertions sur lesquelles s'appuie l'autorité SAML pour produire la décision d'autorisation. Il a le type complexe **EvidenceType**. Il contient un mélange d'un ou plusieurs des éléments suivants:

– <AssertionIDRef> [Tout nombre]

Spécifie une assertion par référence à la valeur de l'attribut d'identifiant ID de l'assertion.

– <AssertionURIRef> [Tout nombre]

Spécifie une assertion au moyen d'une référence d'URI.

– <Assertion> [Tout nombre]

Spécifie une assertion par valeur.

– <EncryptedAssertion> [Tout nombre]

Spécifie une assertion chiffrée par valeur.

Fournir une assertion comme évidence peut affecter l'accord de confiance entre le consommateur d'assertions SAML et l'autorité SAML qui prend la décision d'autorisation. Par exemple, dans le cas où le consommateur d'assertions SAML a présenté une assertion à l'autorité SAML dans une demande, l'autorité SAML peut utiliser cette assertion comme évidence dans une prise de décision d'autorisation sans souscrire à la validité de l'assertion de l'élément <Evidence> envers le consommateur d'assertions ou envers toute autre tierce partie.

Le fragment de schéma suivant définit l'élément <Evidence> et son type complexe **EvidenceType**:

```
<element name="Evidence" type="saml:EvidenceType"/>
<complexType name="EvidenceType">
      <choice maxOccurs="unbounded">
            <element ref="saml:AssertionIDRef"/>
            <element ref="saml:AssertionURIRef"/>
            <element ref="saml:Assertion"/>
            <element ref="saml:EncryptedAssertion"/>
      </choice>
</complexType>
```

8.2 Protocoles SAML

Les messages de protocole SAML peuvent être générés et échangés en utilisant divers protocoles. Les liens SAML au § 10 décrivent des moyens spécifiques pour le transport des messages de protocole en utilisant des protocoles de transport existants et largement déployés. Le profil SAML au § 11 décrit un certain nombre d'applications des protocoles définis dans le présent paragraphe ainsi que des règles de traitement, de restrictions, et d'exigences supplémentaires qui facilitent l'interopérabilité.

Les messages de demande et réponse spécifiques de SAML dérivent des types communs. Le demandeur envoie un élément dérivé de **RequestAbstractType** à un répondant SAML, et le répondant génère un élément conforme à **StatusResponseType** ou en dérivant, comme indiqué à la Figure 8-1.

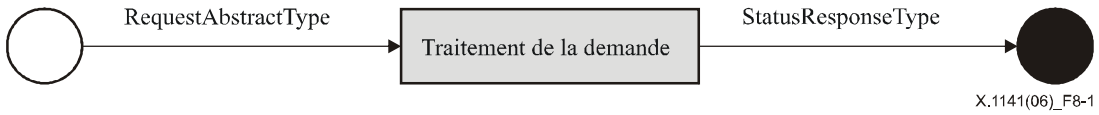

Figure 8-1/X.1141 – Protocole de demande/réponse SAML

Dans certains cas, quand les profils le permettent, une réponse SAML peut être générée et envoyée sans que le répondant ait reçu une demande correspondante.

Les protocoles définis par SAML réalisent les actions suivantes:

- retourner une ou plusieurs assertions demandées. Cela peut survenir en réponse à une demande directe pour des assertions spécifiques ou à une interrogation pour des assertions qui satisfont à un critère particulier;
- effectuer l'authentification à la demande et retourner l'assertion correspondante;
- enregistrer un identifiant de nom ou terminer un enregistrement de nom à la demande;
- restituer un message de protocole qui avait été demandé au moyen d'un artifice;

– effectuer à la demande une sortie de session presque simultanée d'une collection de sessions en rapport entre elles ("sortie unique");

– fournir une transposition d'identifiant de nom à la demande.

Dans le présent paragraphe, les descriptions textuelles des éléments et types dans l'espace de nom de protocole SAML ne sont pas montrées avec le préfixe d'espace de nom conventionnel `samlp:`. Dans un souci de clarté, les descriptions textuelles des éléments et types dans les espaces de nom d'assertion SAML sont indiquées avec le préfixe d'espace de nom conventionnel `saml:`.

8.2.1 En-tête de schéma et déclarations d'espace de nom

Le fragment de schéma suivant définit les espaces de nom XML et autres informations d'en-tête pour le schéma de protocole:

```
<schema
    targetNamespace="urn:oasis:names:tc:SAML:2.0:protocol"
    xmlns="http://www.w3.org/2001/XMLSchema"
    xmlns:samlp="urn:oasis:names:tc:SAML:2.0:protocol"
    xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion"
    xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
    elementFormDefault="unqualified"
    attributeFormDefault="unqualified"
    blockDefault="substitution"
    version="2.0">
    <import namespace="urn:oasis:names:tc:SAML:2.0:assertion"
        schemaLocation="saml-schema-assertion-2.0.xsd"/>
    <import namespace="http://www.w3.org/2000/09/xmldsig#"
        schemaLocation="http://www.w3.org/TR/2002/REC-xmldsig-core-
20020212/xmldsig-core-schema.xsd"/>
    <annotation>
        <documentation>
            Document identifier: saml-schema-protocol-2.0
            Location: http://docs.oasis-open.org/security/saml/v2.0/
            Revision history:
            V1.0 (November, 2002):
              Initial Standard Schema.
            V1.1 (September, 2003):
              Updates within the same V1.0 namespace.
            V2.0 (March, 2005):
              New protocol schema based in a SAML V2.0 namespace.
     </documentation>
    </annotation>
…
</schema>
```

8.2.2 Demandes et réponses

Les paragraphes suivants définissent les constructions SAML et les exigences de base qui sous-tendent tous les messages de demande et réponse utilisés dans les protocoles SAML.

8.2.2.1 Type complexe RequestAbstractType

Toutes les demandes SAML sont de types qui sont dérivés du type complexe abstrait **RequestAbstractType**. Ce type définit des attributs et éléments communs qui sont associés à toutes les demandes SAML:

NOTE – L'élément <RespondWith> a été retiré de **RequestAbstractType** pour la version 2.0 de SAML.

– `ID` [Exigé]

  Identifiant pour la demande. Il est du type **xs:ID** et doit suivre les exigences spécifiées au § 7.4 pour l'unicité de l'identifiant. Les valeurs de l'attribut d'identifiant dans une demande et de l'attribut InResponseTo dans la réponse correspondante doivent coincider.

– `Version` [Exigé]

  Version de cette demande. L'identifiant de cette version de SAML défini dans la présente Recommandation est "2.0".

– `IssueInstant` [Exigé]

  Instant de production de la demande. La valeur horaire est codée en UTC, comme décrit au § 7.3.

– `Destination` [Facultatif]

  Référence d'URI qui indique l'adresse à laquelle cette demande a été envoyée. Elle est utile pour empêcher la retransmission malveillante de demandes à des receveurs imprévus, protection qui est exigée par certains liens de protocoles. Si elle est présente, le receveur réel doit vérifier que la référence d'URI identifie la localisation à laquelle le message a été reçu. Si elle ne l'est pas, la demande doit être éliminée. Certains liens de protocole peuvent exiger l'utilisation de cet attribut (voir le § 10).

– `Consent` [Facultatif]

  Indique si (et sous quelles conditions) le consentement à l'envoi de cette demande a été obtenu ou non de la part d'un principal. Voir au § 8.7.4 quelques références d'URI qui peuvent être utilisées comme valeur de l'attribut `Consent` et leurs descriptions associées. Si aucune valeur de `Consent` n'est fournie, l'identifiant `urn:oasis:names:tc:SAML:2.0:consent:unspecified` est en vigueur.

– `<saml:Issuer>` [Facultatif]

  Identifie l'entité qui a généré le message de demande.

– `<ds:Signature>` [Facultatif]

  Signature XML qui authentifie le demandeur et fournit l'intégrité de message, comme décrit ci-dessous et au paragraphe 8.4.

– `<Extensions>` [Facultatif]

  Ce point d'extension contient des éléments d'extension de message de protocole facultatifs qui sont acceptés entre les parties à la communication. Aucun schéma d'extension n'est exigé pour utiliser ce point d'extension, et même s'il en est fourni un, le réglage imprécis de validation n'impose aucune exigence que l'extension soit valide. Les éléments d'extension SAML doivent être à espace de nom qualifié dans un espace de nom non défini par SAML.

Selon les exigences des protocoles ou profils particuliers, un demandeur SAML peut souvent avoir besoin de s'authentifier, et l'intégrité de message peut souvent être requise. L'authentification et l'intégrité de message peuvent être fournies par des mécanismes apportés par la liaison de protocole (voir le § 10). La demande SAML peut être signée, ce qui apporte à la fois l'authentification du demandeur et l'intégrité du message.

Si une telle signature est utilisée, l'élément `<ds:Signature>` doit alors être présent, et le répondant SAML doit vérifier que la signature est valide (c'est-à-dire que le message n'a pas été altéré) conformément à Signature du W3C. Si elle est non valide, le répondant ne doit pas s'appuyer sur le contenu de la demande et devrait répondre par une erreur. Si elle est valide, le répondant devrait alors évaluer la signature pour déterminer l'identité et la pertinence du signataire et peut continuer à traiter la demande ou répondre par une erreur (si la demande est non valide pour quelque autre raison).

Si un attribut `Consent` est inclus et si la valeur indique qu'une forme de consentement du principal a été obtenue, la demande devrait alors être signée.

Si un répondant SAML estime qu'une demande est non valide conformément à la syntaxe ou aux règles de traitement SAML, si il répond, il doit retourner un message de réponse SAML avec un élément `<StatusCode>` de valeur `urn:oasis:names:tc:SAML:2.0:status:Requester`. Dans certains cas, par exemple durant un soupçon d'attaque de déni de service, il peut être plus sûr de ne pas répondre du tout.

Le fragment de schéma suivant définit le type complexe **RequestAbstractType**:

```
<complexType name="RequestAbstractType" abstract="true">
       <sequence>
              <element ref="saml:Issuer" minOccurs="0"/>
              <element ref="ds:Signature" minOccurs="0"/>
              <element ref="samlp:Extensions" minOccurs="0"/>
       </sequence>
       <attribute name="ID" type="ID" use="required"/>
       <attribute name="Version" type="string" use="required"/>
       <attribute name="IssueInstant" type="dateTime" use="required"/>
       <attribute name="Destination" type="anyURI" use="optional"/>
       <attribute name="Consent" type="anyURI" use="optional"/>
</complexType>
<element name="Extensions" type="samlp:ExtensionsType"/>
<complexType name="ExtensionsType">
```

```
      <sequence>
            <any namespace="##other" processContents="lax"
maxOccurs="unbounded"/>
      </sequence>
</complexType>
```

8.2.2.2 Type complexe type StatusResponseType

Toutes les réponses SAML sont de types dérivés du type complexe **StatusResponseType**. Ce type définit des attributs et éléments communs qui sont associés à toutes les réponses SAML:

– `ID` [Exigé]

Identifiant de la réponse. Il est du type **xs:ID**, et doit suivre les exigences spécifiées au paragraphe 7.4 pour l'unicité de l'identifiant.

– `InResponseTo` [Facultatif]

Référence de l'identifiant de la demande à laquelle correspond la réponse, s'il en est. Si la réponse n'est pas générée en réponse à une demande, ou si la valeur d'attribut d'identifiant d'une demande ne peut être déterminée (par exemple, la demande est mal formée), cet attribut ne doit alors pas être présent. Autrement, il doit être présent et sa valeur doit correspondre à la valeur de l'attribut d'identifiant de la demande correspondante.

– `Version` [Exigé]

Version de cette réponse. L'identifiant de la version de SAML définie dans la présente Recommandation est "2.0".

– `IssueInstant` [Exigé]

Instant de production de la réponse. La valeur horaire est codée en UTC, comme décrit au paragraphe 7.3.

– `Destination` [Facultatif]

Référence d'URI qui indique l'adresse à laquelle cette réponse a été envoyée. Elle est utile pour empêcher la retransmission malveillante de demandes à des receveurs imprévus, protection qui est exigée par certains liens de protocoles. Si elle est présente, le receveur réel doit vérifier que la référence d'URI identifie la localisation à laquelle le message a été reçu. Si elle ne l'est pas, la réponse doit être éliminée. Certains liens de protocole peuvent exiger l'utilisation de cet attribut (voir le § 10).

– `Consent` [Facultatif]

Indique si (et sous quelles conditions) le consentement à l'envoi de cette réponse a été obtenu ou non de la part d'un principal. Voir au § 8.7.4 quelques références d'URI qui peuvent être utilisées comme valeur de l'attribut `Consent` et leurs descriptions associées. Si aucune valeur de `Consent` n'est fournie, l'identifiant `urn:oasis:names:tc:SAML:2.0:consent:unspecified` (voir au § 8.7.4) est en vigueur.

– `<saml:Issuer>` [Facultatif]

Identifie l'entité qui a généré le message de réponse. (Pour des informations complémentaires sur cet élément, voir au § 8.1.2.5).

– `<ds:Signature>` [Facultatif]

Signature XML qui authentifie le répondant et assure l'intégrité du message, comme décrit ci-dessous et au § 8.4.

– `<Extensions>` [Facultatif]

Ce point d'extension contient des éléments d'extension de message de protocole facultatifs qui ont été acceptés entre les parties à la communication. Aucun schéma d'extension n'est requis pour l'utilisation de ce point d'extension, et même s'il en est fourni un, le réglage souple de validation n'impose d'aucune façon que l'extension soit valide. Les éléments d'extension SAML doivent avoir un espace de nom qualifié dans un espace de nom non défini par SAML.

– `<Status>` [Exigé]

Code représentant l'état de la demande correspondante.

Selon les exigences des protocoles ou profils particuliers, un répondant SAML peut souvent avoir besoin de s'authentifier, et l'intégrité du message peut souvent être requise. L'authentification et l'intégrité du message peuvent être fournies par des mécanismes apportés par la liaison de protocole. La réponse SAML peut être signée, ce qui procure à la fois l'authentification du répondant et l'intégrité du message.

Si une telle signature est utilisée, l'élément <ds:Signature> doit alors être présent, et le demandeur SAML qui reçoit la réponse doit vérifier que la signature est valide (c'est-à-dire que le message n'a pas été altéré), Signature XML du W3C. Si elle est non valide, le demandeur ne doit pas s'appuyer sur le contenu de la réponse et devrait la traiter comme une erreur. Si elle est valide, le demandeur devrait évaluer la signature pour déterminer l'identité et la pertinence du signataire et peut continuer à traiter la réponse comme il l'estime approprié.

Si un attribut Consent est inclus et si sa valeur indique qu'une certaine forme de consentement de principal a été obtenu, la réponse devrait alors être signée.

Le fragment de schéma suivant définit le type complexe **StatusResponseType**:

```
<complexType name="StatusResponseType">
      <sequence>
            <element ref="saml:Issuer" minOccurs="0"/>
            <element ref="ds:Signature" minOccurs="0"/>
            <element ref="samlp:Extensions" minOccurs="0"/>
            <element ref="samlp:Status"/>
      </sequence>
      <attribute name="ID" type="ID" use="required"/>
      <attribute name="InResponseTo" type="NCName" use="optional"/>
      <attribute name="Version" type="string" use="required"/>
      <attribute name="IssueInstant" type="dateTime" use="required"/>
      <attribute name="Destination" type="anyURI" use="optional"/>
      <attribute name="Consent" type="anyURI" use="optional"/>
</complexType>
```

1) Élément <Status>

L'élément <Status> contient les éléments suivants:

– <StatusCode> [Exigé]

  Code représentant l'état de l'activité effectuée en réponse à la demande correspondante.

– <StatusMessage> [Facultatif]

  Message qui peut être retourné à un opérateur.

– <StatusDetail> [Facultatif]

  Informations supplémentaires concernant l'état de la demande.

Le fragment de schéma suivant définit l'élément <Status> et son type complexe **StatusType**:

```
<element name="Status" type="samlp:StatusType"/>
<complexType name="StatusType">
    <sequence>
       <element ref="samlp:StatusCode"/>
       <element ref="samlp:StatusMessage" minOccurs="0"/>
       <element ref="samlp:StatusDetail" minOccurs="0"/>
    </sequence>
</complexType>
```

2) Élément <StatusCode>

L'élément <StatusCode> spécifie un code ou un ensemble de codes enchassés représentant l'état de la demande correspondante. L'élément <StatusCode> a l'élément et attribut suivants:

– Valeur [Exigé]

  Valeur de code d'état. Cet attribut contient une référence d'URI. La valeur de l'élément <StatusCode> le plus élevé doit être tiré de la liste de niveau supérieur fournie dans le présent paragraphe.

– <StatusCode> [Facultatif]

  Code d'état subordonné qui fournit plus d'informations spécifiques sur une condition d'erreur. Les répondants peuvent omettre les codes d'état subordonnés afin d'empêcher des attaques qui cherchent à collecter des informations supplémentaires en présentant intentionnellement des demandes erronées.

Les valeurs de <StatusCode> de haut niveau permises sont les suivantes:

```
urn:oasis:names:tc:SAML:2.0:status:Success
```

La demande a réussi. Des informations supplémentaires peuvent être retournées dans les éléments <StatusMessage> et/ou <StatusDetail>.

```
urn:oasis:names:tc:SAML:2.0:status:Requester
```

La demande n'a pas pu être effectuée à cause d'une erreur de la part du demandeur.

```
urn:oasis:names:tc:SAML:2.0:status:Responder
```

La demande n'a pas pu être effectuée à cause d'une erreur de la part du répondant SAML ou de l'autorité SAML.

```
urn:oasis:names:tc:SAML:2.0:status:VersionMismatch
```

Le répondant SAML n'a pas pu traiter la demande parce que la version du message de demande était incorrecte.

Les codes d'état de second niveau suivants sont référencés à divers endroits dans la présente Recommandation. Des codes d'état de second niveau supplémentaires peuvent être définis dans de futures versions de la Recommandation SAML. Les entités système ont toute liberté pour définir plus de codes d'état spécifiques en définissant les références d'URI appropriées.

```
urn:oasis:names:tc:SAML:2.0:status:AuthnFailed
```

Le fournisseur répondant n'a pas été capable d'authentifier avec succès le principal.

```
urn:oasis:names:tc:SAML:2.0:status:InvalidAttrNameOrValue
```

Un contenu inattendu ou non valide s'est trouvé au sein d'un élément <saml:Attribute> ou <saml:AttributeValue>.

```
urn:oasis:names:tc:SAML:2.0:status:InvalidNameIDPolicy
```

Le fournisseur répondant ne peut ou ne veut pas prendre en charge la politique d'identifiant de nom demandée.

```
urn:oasis:names:tc:SAML:2.0:status:NoAuthnContext
```

Le répondant ne peut pas satisfaire aux exigences de contexte d'authentification spécifiées.

```
urn:oasis:names:tc:SAML:2.0:status:NoAvailableIDP
```

Utilisé par un intermédiaire pour indiquer qu'aucun des éléments <Loc> de fournisseur d'identité accepté dans <IDPList> ne peut être résolu ou qu'aucun des fournisseur d'identités acceptés n'est disponible.

```
urn:oasis:names:tc:SAML:2.0:status:NoPassive
```

Indique que le fournisseur répondant ne peut pas authentifier passivement le principal, comme il lui a été demandé.

```
urn:oasis:names:tc:SAML:2.0:status:NoSupportedIDP
```

Utilisé par un intermédiaire pour indiquer qu'aucun des fournisseurs d'identité dans une <IDPList> n'est accepté par l'intermédiaire.

```
urn:oasis:names:tc:SAML:2.0:status:PartialLogout
```

Utilisé par une autorité de session pour indiquer à un participant à une session qu'il n'a pas été capable de propager l'ouverture de session à tous les autres participants à la session.

```
urn:oasis:names:tc:SAML:2.0:status:ProxyCountExceeded
```

Indique qu'un fournisseur répondant ne peut pas authentifier directement le principal et qu'il ne lui est pas permis de remandater la demande.

```
urn:oasis:names:tc:SAML:2.0:status:RequestDenied
```

Le répondant SAML ou l'autorité SAML est capable de traiter la demande mais a choisi de ne pas répondre. Ce code d'état peut être utilisé lorsqu'il y a un souci à propos du contexte de sécurité du message de demande ou de la séquence de messages de demande reçus d'un demandeur particulier.

```
urn:oasis:names:tc:SAML:2.0:status:RequestUnsupported
```

Le répondant SAML ou l'autorité SAML ne prend pas la demande en charge.

```
urn:oasis:names:tc:SAML:2.0:status:RequestVersionDeprecated
```

Le répondant SAML ne peut traiter aucune demande avec la version de protocole spécifiée dans la demande.

```
urn:oasis:names:tc:SAML:2.0:status:RequestVersionTooHigh
```

Le répondant SAML ne peut pas traiter la demande parce que la version de protocole spécifiée dans le message de demande est une mise à jour majeure de la plus haute version de protocole acceptée par le répondant.

```
urn:oasis:names:tc:SAML:2.0:status:RequestVersionTooLow
```

Le répondant SAML ne peut pas traiter la demande parce que la version de protocole spécifiée dans le message de demande est trop ancienne.

```
urn:oasis:names:tc:SAML:2.0:status:ResourceNotRecognized
```

La valeur de ressource fournie dans le message de demande est non valide ou non reconnue.

```
urn:oasis:names:tc:SAML:2.0:status:TooManyResponses
```

Le message de réponse contiendrait plus d'éléments que ce que le répondant SAML est capable de retourner.

```
urn:oasis:names:tc:SAML:2.0:status:UnknownAttrProfile
```

Une entité qui n'a aucune connaissance d'un profil d'attribut particulier a été mise en présence d'un attribut tiré de ce profil.

```
urn:oasis:names:tc:SAML:2.0:status:UnknownPrincipal
```

Le fournisseur répondant ne reconnait pas le principal spécifié ou impliqué par la demande.

```
urn:oasis:names:tc:SAML:2.0:status:UnsupportedBinding
```

Le répondant SAML ne peut pas satisfaire correctement la demande en utilisant la liaison de protocole spécifiée dans la demande.

Le fragment de schéma suivant définit l'élément `<StatusCode>` et son type complexe **StatusCodeType**:

```
<element name="StatusCode" type="samlp:StatusCodeType"/>
<complexType name="StatusCodeType">
    <sequence>
       <element ref="samlp:StatusCode" minOccurs="0"/>
    </sequence>
    <attribute name="Value" type="anyURI" use="required"/>
</complexType>
```

3) Élément <StatusMessage>

L'élément `<StatusMessage>` spécifie un message qui peut être retourné à un opérateur:

Le fragment de schéma suivant définit l'élément `<StatusMessage>`:

```
<element name="StatusMessage" type="string"/>
```

4) Élément <StatusDetail>

L'élément `<StatusDetail>` peut être utilisé pour spécifier des informations supplémentaires concernant l'état de la demande. Les informations supplémentaires consistent en zéro, un ou plusieurs éléments tirés de tout espace de nom, sans exigence qu'un schéma soit présent ou d'une validation de schéma du contenu de `<StatusDetail>`.

Le fragment de schéma suivant définit l'élément `<StatusDetail>` et son type complexe **StatusDetailType**:

```
<element name="StatusDetail" type="samlp:StatusDetailType"/>
<complexType name="StatusDetailType">
    <sequence>
       <any namespace="##any" processContents="lax" minOccurs="0"
       maxOccurs="unbounded"/>
    </sequence>
</complexType>
```

8.2.3 Interrogation d'assertion et protocole de demande

Le présent paragraphe définit les messages et règles de traitement pour demander les assertions existantes par référence ou interrogation des assertions par sujet et type de déclaration.

8.2.3.1 Élément <AssertionIDRequest>

Si le demandeur connait l'identifiant unique d'une ou plusieurs assertions, l'élément de message `<AssertionIDRequest>` peut être utilisé pour demander qu'il soit retourné dans un message `<Response>`. L'élément `<saml:AssertionIDRef>` est utilisé pour spécifier chaque assertion à retourner.

Le fragment de schéma suivant définit l'élément <AssertionIDRequest>:

```
<element name="AssertionIDRequest" type="samlp:AssertionIDRequestType"/>
<complexType name="AssertionIDRequestType">
      <complexContent>
            <extension base="samlp:RequestAbstractType">
                  <sequence>
                        <element ref="saml:AssertionIDRef"
maxOccurs="unbounded"/>
                  </sequence>
            </extension>
      </complexContent>
</complexType>
```

8.2.3.2 Interrogations

Les paragraphes suivants définissent les messages de demande d'interrogation SAML.

8.2.3.2.1 Élément <SubjectQuery>

L'élément de message <SubjectQuery> est un point d'extension qui permet à de nouvelles interrogations SAML d'être définies pour spécifier un seul sujet SAML. Son type complexe **SubjectQueryAbstractType** est abstrait et il n'est donc utilisable que comme base d'un type dérivé. **SubjectQueryAbstractType** ajoute l'élément <saml:Subject> (défini au paragraphe 8.1.4) à **RequestAbstractType**.

Le fragment de schéma suivant définit l'élément <SubjectQuery> et son type complexe **SubjectQueryAbstractType**:

```
<element name="SubjectQuery" type="samlp:SubjectQueryAbstractType"/>
<complexType name="SubjectQueryAbstractType" abstract="true">
     <complexContent>
            <extension base="samlp:RequestAbstractType">
                  <sequence>
                        <element ref="saml:Subject"/>
                  </sequence>
            </extension>
     </complexContent>
</complexType>
```

8.2.3.2.2 Élément <AuthnQuery>

L'élément de message <AuthnQuery> est utilisé pour faire l'interrogation "Quelles assertions contenant des déclarations d'authentification sont disponibles pour ce sujet ?" Une <Response> réussie contiendra une ou plusieurs assertions contenant des déclarations d'authentification.

Le message <AuthnQuery> ne doit pas être utilisé comme demande de nouvelle authentification en utilisant les accréditifs fournis dans la demande. <AuthnQuery> est une demande de déclarations sur des actes d'authentification qui sont survenus dans une interaction précédente entre le sujet indiqué et l'autorité d'authentification.

Cet élément est de type **AuthnQueryType**, qui étend **SubjectQueryAbstractType** avec l'ajout de l'élément et de l'attribut suivants:

– SessionIndex [Facultatif]

  S'il est présent, spécifie un filtre pour les réponses possibles. Une telle interrogation pose la question "Quelles assertions contenant des déclarations d'authentification avez-vous pour ce sujet dans le contexte des informations de session fournies ?"

– <RequestedAuthnContext> [Facultatif]

  S'il est présent, spécifie un filtre pour les réponses possibles. Une telle interrogation pose la question "Quelles assertions contenant des déclarations d'authentification avez-vous pour ce sujet qui satisfont aux exigences de contexte d'authentification dans cet élément ?"

En réponse à une interrogation d'authentification, une autorité SAML retourne les assertions avec les déclarations d'authentification comme suit:

- les règles données au § 8.2.3.4 pour la correspondance avec l'élément <Subject> de l'interrogation identifient les assertions qui peuvent être retournées;
- si l'attribut SessionIndex est présent dans l'interrogation, au moins un élément <AuthnStatement> de l'ensemble des assertions retournées doit contenir un attribut SessionIndex qui corresponde à

l'attribut SessionIndex de l'interrogation. Il est facultatif de retourner l'ensemble complet de telles assertions correspondantes dans la réponse;

- si l'élément <RequestedAuthnContext> est présent dans l'interrogation, au moins un élément <AuthnStatement> de l'ensemble des assertions retournées doit contenir un élément <AuthnContext> qui satisfait à l'élément dans l'interrogation. Il est facultatif de retourner l'ensemble complet de telles assertions correspondantes dans la réponse.

Le fragment de schéma suivant définit l'élément <AuthnQuery> et son type complexe **AuthnQueryType**:

```
<element name="AuthnQuery" type="samlp:AuthnQueryType"/>
<complexType name="AuthnQueryType">
       <complexContent>
              <extension base="samlp:SubjectQueryAbstractType">
                     <sequence>
                            <element ref="samlp:RequestedAuthnContext"
minOccurs="0"/>
                     </sequence>
                     <attribute name="SessionIndex" type="string"
use="optional"/>
              </extension>
       </complexContent>
</complexType>
```

1) Élément <RequestedAuthnContext>

L'élément <RequestedAuthnContext> spécifie les exigences de contexte d'authentification des déclarations d'authentification retournées dans les réponses à une demande ou une interrogation. Son type complexe **RequestedAuthnContextType** définit les éléments et attributs suivants:

– <saml:AuthnContextClassRef> ou <saml:AuthnContextDeclRef> [Un ou plusieurs]

Spécifie une ou plusieurs références d'URI identifiant des classes ou déclarations de contexte d'authentification. Ces éléments sont définis au § 8.1.7.2.2. Pour des informations complémentaires sur les classes de contexte d'authentification, voir le § 12.

– Comparison [Facultatif]

Spécifie la méthode de comparaison utilisée pour évaluer les classes ou déclarations d'un contexte demandé, "exact", "minimum", "maximum", ou "better" *(meilleur que)*. La valeur par défaut est "exact".

On peut utiliser un ensemble de références de classe ou un ensemble de références de déclaration. L'ensemble des références fournies doit être évalué comme un ensemble ordonné, où le premier élément est la classe ou déclaration de contexte d'authentification préférée. Si aucune des classes ou déclarations spécifiées ne peut être satisfaite conformément aux règles ci-dessous, le répondant doit alors retourner un message <Response> avec un <StatusCode> de second niveau de urn:oasis:names:tc:SAML:2.0:status:NoAuthnContext.

Si Comparison est mis à "exact" ou est omis, le contexte d'authentification résultant dans la déclaration d'authentification doit alors être la correspondance exacte d'au moins un des contextes d'authentification spécifiés.

Si Comparison est mis à "minimum", le contexte d'authentification résultant dans la déclaration d'authentification doit alors être au moins aussi fort (selon la supposition du répondant) qu'un de ceux des contextes d'authentification spécifiés.

Si Comparison est mis à "better", le contexte d'authentification résultant dans la déclaration d'authentification doit alors être plus fort (selon la supposition du répondant) que tous ceux des contextes d'authentification spécifiés.

Si Comparison est mis à "maximum", le contexte d'authentification résultant dans la déclaration d'authentification doit alors être aussi fort que possible (selon la supposition du répondant) sans excéder la force d'au moins un des contextes d'authentification spécifiés.

Le fragment de schéma suivant définit l'élément <RequestedAuthnContext> et son type complexe **RequestedAuthnContextType**:

```
<element name="RequestedAuthnContext" type="samlp:RequestedAuthnContextType"/>
<complexType name="RequestedAuthnContextType">
  <choice>
         <element ref="saml:AuthnContextClassRef" maxOccurs="unbounded"/>
         <element ref="saml:AuthnContextDeclRef" maxOccurs="unbounded"/>
  </choice>
```

```
 <attribute name="Comparison" type="samlp:AuthnContextComparisonType"
use="optional"/>
</complexType>
<simpleType name="AuthnContextComparisonType">
 <restriction base="string">
       <enumeration value="exact"/>
       <enumeration value="minimum"/>
       <enumeration value="maximum"/>
       <enumeration value="better"/>
 </restriction>
</simpleType>
```

8.2.3.2.3 Élément <AttributeQuery>

L'élément `<AttributeQuery>` sert à faire l'interrogation "Retourner les attributs demandés pour ce sujet." Une réponse de succès sera sous la forme d'assertions contenant des déclarations d'attributs, selon la quantité admise par la politique. Cet élément est du type **AttributeQueryType**, qui étend **SubjectQueryAbstractType** avec l'ajout de l'élément suivant:

– `<saml:Attribute>` [Tout nombre]

Chaque élément `<saml:Attribute>` spécifie un attribut dont la ou les valeurs sont à retourner. Si aucun attribut n'est spécifié, il indique que tous les attributs admis par la politique sont demandés. Si un élément `<saml:Attribute>` donné contient un ou plusieurs éléments `<saml:AttributeValue>`, si cet attribut est retourné dans la réponse, il ne doit pas contenir de valeurs qui ne soient pas égales aux valeurs spécifiées dans l'interrogation. En l'absence de règles d'égalité spécifiées par des profils ou attributs particuliers, l'égalité est définie comme une représentation XML identique de la valeur. Pour des informations complémentaires sur `<saml:Attribute>`, voir au § 8.6.

Une seule interrogation ne doit pas contenir deux éléments `<saml:Attribute>` avec les mêmes valeurs `Name` et `NameFormat` (c'est-à-dire qu'un attribut donné doit être nommé une seule fois dans une interrogation).

En réponse à une interrogation d'attribut, une autorité SAML retourne des assertions avec des déclarations d'attribut comme suit:

- les règles données au § 8.2.3.4 pour la correspondance avec l'élément `<Subject>` de l'interrogation identifient les assertions qui peuvent être retournées;
- si un élément `<Attribute>` quelconque est présent dans l'interrogation, il contraint/filtre les attributs et facultativement les valeurs retournées comme noté ci-dessus;
- les attributs et valeurs retournées peuvent aussi être contraintes par des considérations de politique spécifiques de l'application.

Les codes d'état de second niveau `urn:oasis:names:tc:SAML:2.0:status:UnknownAttrProfile` et `urn:oasis:names:tc:SAML:2.0:status:InvalidAttrNameOrValue` peuvent être utilisés pour indiquer des problèmes avec l'interprétation des informations d'attribut ou de valeur dans une interrogation.

Le fragment de schéma suivant définit l'élément `<AttributeQuery>` et son type complexe **AttributeQueryType**:

```
<element name="AttributeQuery" type="samlp:AttributeQueryType"/>
<complexType name="AttributeQueryType">
       <complexContent>
              <extension base="samlp:SubjectQueryAbstractType">
                     <sequence>
                            <element ref="saml:Attribute" minOccurs="0"
maxOccurs="unbounded"/>
                     </sequence>
              </extension>
       </complexContent>
</complexType>
```

8.2.3.2.4 Élément <AuthzDecisionQuery>

L'élément `<AuthzDecisionQuery>` est utilisé pour faire l'interrogation "Ces actions sur cette ressource devraient-elles être admises pour ce sujet, étant donné cette évidence ?" Une réponse de succès sera sous la forme d'assertions contenant des déclarations de décision d'autorisation.

NOTE – La caractéristique `<AuthzDecisionQuery>` a été gelée dans SAML V2.0, sans améliorations futures prévues. Les utilisateurs qui ont besoin de fonctionnalités supplémentaires pourraient regarder dans le langage de balisage de contrôle d'accès extensible (voir la Rec. UIT-T X.1142), qui offre des caractéristiques de décision d'autorisation améliorées.

Cet élément est du type **AuthzDecisionQueryType**, qui étend **SubjectQueryAbstractType** avec l'ajout des éléments et attributs suivants:

– Resource [Exigé]

Référence d'URI indiquant la ressource pour laquelle l'autorisation est demandée.

– <saml:Action> [Une ou plusieurs]

Les actions pour lesquelles l'autorisation est demandée. Pour plus d'informations sur cet élément, voir au paragraphe 8.1.7.4.2.

– <saml:Evidence> [Facultatif]

Ensemble d'assertions sur lesquelles l'autorité SAML peut s'appuyer en prenant sa décision d'autorisation. Pour plus d'informations sur cet élément, voir au § 8.1.7.4.3.

En réponse à une interrogation de décision d'autorisation, une autorité SAML retourne des assertions avec des déclarations de décision d'autorisation comme suit:

- les règles données au § 8.2.3.4 pour la correspondance avec l'élément `<Subject>` de l'interrogation identifient les assertions qui peuvent être retournées.

Le fragment de schéma suivant définit l'élément `<AuthzDecisionQuery>` et son type complexe **AuthzDecisionQueryType**:

```
<element name="AuthzDecisionQuery" type="samlp:AuthzDecisionQueryType"/>
<complexType name="AuthzDecisionQueryType">
 <complexContent>
       <extension base="samlp:SubjectQueryAbstractType">
             <sequence>
                   <element ref="saml:Action" maxOccurs="unbounded"/>
                   <element ref="saml:Evidence" minOccurs="0"/>
             </sequence>
             <attribute name="Resource" type="anyURI" use="required"/>
       </extension>
 </complexContent>
</complexType>
```

8.2.3.3 Élément <Response>

L'élément de message `<Response>` est utilisé lorsqu'une réponse consiste en une liste de zéro, une ou plusieurs assertions qui satisfont la demande. Il a le type complexe **ResponseType**, qui étend **StatusResponseType** et ajoute les éléments suivants:

– `<saml:Assertion>` ou `<saml:EncryptedAssertion>` [Tout nombre]

Spécifie une assertion par valeur, ou facultativement une assertion chiffrée par valeur. Voir au § 8.1.3.3 des informations complémentaires sur ces éléments.

Le fragment de schéma suivant définit l'élément <Response> et son type complexe **ResponseType**:

```
<element name="Response" type="samlp:ResponseType"/>
<complexType name="ResponseType">
       <complexContent>
             <extension base="samlp:StatusResponseType">
                   <choice minOccurs="0" maxOccurs="unbounded">
                         <element ref="saml:Assertion"/>
                         <element ref="saml:EncryptedAssertion"/>
                   </choice>
             </extension>
       </complexContent>
</complexType>
```

8.2.3.4 Règles de traitement

En réponse à un message d'interrogation défini par SAML, chaque assertion retournée par une autorité SAML doit contenir un élément `<saml:Subject>` qui **correspond fortement** à l'élément `<saml:Subject>` trouvé dans l'interrogation.

L'élément <saml:Subject> S1 correspond fortement à S2 si et seulement si les deux conditions suivantes s'appliquent toutes deux:

- si S2 inclut un élément identifiant (<BaseID>, <NameID>, ou <EncryptedID>), alors S1 doit inclure un élément identifiant identique, mais l'élément peut être chiffré (ou pas) en S1 ou en S2. En d'autres termes, la forme déchiffrée de l'identifiant doit être identique en S1 et en S2. "Identique" signifie que les contenus de l'élément identifiant et des valeurs d'attribut doivent être les mêmes. Un identifiant chiffré sera identique à l'original selon cette définition, une fois déchiffré;
- si S2 inclut un ou plusieurs éléments <saml:SubjectConfirmation>, S1 doit alors inclure au moins un élément <saml:SubjectConfirmation> tel que S1 puisse être confirmé de la manière décrite par au moins un élément <saml:SubjectConfirmation> dans S2.

A titre d'exemple de ce qui est permis et de ce qui ne l'est pas, S1 pourrait contenir un <saml:NameID> avec une valeur de Format particulière, et S2 pourrait contenir un élément <saml:EncryptedID> qui serait le résultat du chiffrement de l'élément <saml:NameID> de S1. Cependant, S1 et S2 ne peuvent pas contenir un élément <saml:NameID> avec des valeurs de Format et de contenu d'élément différentes, même si les deux identifiants sont considérés comme se référant au même principal.

Si l'autorité SAML ne peut pas fournir une assertion avec une déclaration satisfaisant aux contraintes exprimées par une interrogation ou référence d'assertion, l'élément <Response> ne doit pas contenir d'élément <Assertion> et doit inclure un élément <StatusCode> avec la valeur urn:oasis:names:tc:SAML:2.0:status:Success.

Toutes les autres règles de traitement associées aux messages sous-jacents de demande et de réponse doivent être observées.

8.2.4 Protocole de demande d'authentication

Lorsqu'un principal (ou un agent agissant au nom d'un principal) souhaite obtenir des assertions contenant des déclarations d'authentification pour établir un contexte de sécurité chez un ou plusieurs consommateurs d'assertions, il peut utiliser le protocole de demande d'authentification pour envoyer un élément de message <AuthnRequest> à une autorité SAML et demander qu'elle retourne un message <Response> contenant une ou plusieurs semblables assertions. De telles assertions peuvent contenir des déclarations supplémentaires de tout type, mais au moins une assertion doit contenir au moins une déclaration d'authentification. Une autorité SAML qui prend en charge ce protocole est aussi appelée fournisseur d'identité.

A part cette exigence, le contenu spécifique des assertions retournées dépend du profil ou contexte d'utilisation. Le moyen exact par lequel le principal ou l'agent s'authentifie auprès du fournisseur d'identité n'est donc pas spécifié, bien que le moyen d'authentification puisse affecter le contenu de la réponse. Les autres questions se rapportant à la validation des accréditifs d'authentification par le fournisseur d'identité ou aux communications entre le fournisseur d'identité et toute autre entité impliquée dans le processus d'authentification sont également en dehors du domaine d'appplication du présent protocole.

Les descriptions et règles de traitement des paragraphes suivants utilisent des références aux acteurs suivants, dont bon nombre peuvent être une seule et même entité dans un profil d'utilisation particulier:

– Demandeur

  L'entité qui crée la demande d'authentification et à qui la réponse est à retourner.

– Présentateur

  L'entité qui présente la demande au fournisseur d'identité et soit s'authentifie elle-même durant la transmission du message, soit s'appuie sur un contexte de sécurité existant pour établir son identité. S'il n'est pas le demandeur, le présentateur agit comme intermédiaire entre le demandeur et le fournisseur d'identité répondant.

– Sujet demandé

  L'entité à propos de laquelle une ou plusieurs assertions sont demandées.

– Entité témoin

  La ou les entités dont on attend qu'elles soient capables de satisfaire un des éléments <SubjectConfirmation> de la ou des assertions résultantes.

– Consommateur d'assertions

  La ou les entités dont on s'attend qu'elles consomment la ou les assertions pour accomplir un objet défini par le profil ou contexte d'utilisation, généralement pour établir un contexte de sécurité.

– Fournisseur d'identité

L'entité à laquelle le présentateur donne la demande et de laquelle le présentateur reçoit la réponse.

Élément <AuthnRequest>

Pour demander qu'un fournisseur d'identité produise une assertion avec une déclaration d'authentification, un présentateur s'authentifie auprès de ce fournisseur d'identité (ou s'appuie sur un contexte de sécurité existant) et lui envoie un message `<AuthnRequest>` qui décrit les propriétés que doit avoir l'assertion résultante pour satisfaire à son objet. Parmi ces propriétés peuvent figurer les informations qui se rapportent au contenu de l'assertion et/ou aux informations qui se rapportent à la façon dont le message `<Response>` résultant devrait être livré au demandeur. Le processus d'authentification du présentateur peut avoir lieu avant, pendant ou après la livraison initiale du message `<AuthnRequest>`.

Le demandeur peut n'être pas le même que le présentateur de la demande si, par exemple, le demandeur est un consommateur d'assertions qui a l'intention d'utiliser l'assertion résultante pour authentifier ou autoriser le sujet demandé de telle sorte que le consommateur d'assertions puisse décider s'il va fournir un service.

Le message `<AuthnRequest>` devrait être signé ou autrement authentifié et protégé en intégrité par la liaison de protocole utilisée pour délivrer le message.

Ce message a le type complexe **AuthnRequestType**, qui étend **RequestAbstractType** et ajoute les éléments et attributs suivants, dont tous sont en général facultatifs, mais peuvent être nécessaires pour des profils spécifiques:

– `<saml:Subject>` [Facultatif]

Spécifie le sujet demandé de la ou des assertions résultantes. Cela peut inclure un ou plusieurs éléments `<saml:SubjectConfirmation>` pour indiquer comment et/ou par qui les assertions résultantes peuvent être confirmées. Pour des informations complémentaires sur cet élément, voir au § 8.1.4.

S'il est entièrement omis ou si aucun identifiant n'est inclus, le présentateur du message est présumé être le sujet demandé. Si aucun élément `<saml:SubjectConfirmation>` n'est inclus, le présentateur est alors présumé être la seule entité témoin requise et la méthode est impliquée par le profil d'utilisation et/ou les politiques du fournisseur d'identité.

– `<NameIDPolicy>` [Facultatif]

Spécifie les contraintes pesant sur l'utilisation de l'identifiant de nom pour représenter le sujet demandé. S'il est omis, tout type d'identifiant accepté par le fournisseur d'identité pour le sujet demandé peut alors être utilisé, dans les limites fixées par toute politique spécifique du développement pertinente, par rapport, par exemple, à la confidentialité.

– `<saml:Conditions>` [Facultatif]

Spécifie les conditions SAML que le demandeur s'attend à voir limiter la validité et/ou l'utilisation de la ou des assertions résultantes. Le répondant peut modifier ou augmenter cet ensemble autant qu'il le juge nécessaire. Les informations contenues dans cet élément sont utilisées comme entrées dans le processus de construction de l'assertion, plutôt que comme conditions sur l'utilisation de la demande elle-même. (Pour des informations complémentaires sur cet élément, voir au § 8.1.5.)

– `<RequestedAuthnContext>` [Facultatif]

Spécifie les exigences, s'il en est, que le demandeur impose au contexte d'authentification qui s'applique à l'authentification du présentateur du fournisseur répondant.

– `<Scoping>` [Facultatif]

Spécifie un ensemble de fournisseur d'identités en lesquels le demandeur a confiance pour authentifier le présentateur, ainsi que les limitations et le contexte relatifs au mandatement du message `<AuthnRequest>` au fournisseur d'identités suivant par le répondant.

– `ForceAuthn` [Facultatif]

Valeur booléenne. Si elle est mise à "true" *(vrai)*, le fournisseur d'identité doit authentifier le présentateur directement plutôt que de s'appuyer sur un contexte de sécurité précédent. Si une valeur n'est pas fournie, la valeur par défaut est "false" *(faux)*. Cependant, si `ForceAuthn` et `IsPassive` sont tous deux "true", le fournisseur d'identité ne doit pas authentifier à nouveau le présentateur sauf si les contraintes de `IsPassive` peuvent être satisfaites.

– `IsPassive` [Facultatif]

Valeur booléenne. Si elle est mise à "true", le fournisseur d'identité et l'agent d'utilisateur eux-mêmes ne doivent pas prendre le contrôle de l'interface d'utilisateur à partir du demandeur et interagir avec le présentateur d'une façon perceptible. Si aucune valeur n'est fournie, la valeur par défaut est "false".

– `AssertionConsumerServiceIndex` [Facultatif]

Identifie indirectement la localisation à laquelle le message `<Response>` devrait être retourné au demandeur. Il ne s'applique qu'aux profils dans lesquels le demandeur est différent du présentateur, tels que dans le profil de navigateur de la toile SSO dans la présente Recommandation. Le fournisseur d'identité doit avoir un moyen de confiance pour transposer la valeur d'indice dans l'attribut à une localisation associée au demandeur. Le paragraphe 9 donne un mécanisme possible. S'il est omis, le fournisseur d'identité doit retourner le message `<Response>` à la localisation par défaut associée au demandeur pour le profil d'utilisation. Si l'indice spécifié est non valide, le fournisseur d'identité peut retourner un message `<Response>` d'erreur ou il peut utiliser la localisation par défaut. Cet attribut s'exclut mutuellement avec les attributs `AssertionConsumerServiceURL` et `ProtocolBinding`.

– `AssertionConsumerServiceURL` [Facultatif]

Spécifie par valeur la localisation à laquelle le message `<Response>` doit être retourné au demandeur. Le répondant doit s'assurer par un moyen quelconque que la valeur spécifiée est en fait associée au demandeur. Le paragraphe 9 donne un mécanisme possible; signer le message `<AuthnRequest>` d'enveloppe en est un autre. Cet attribut s'exclut mutuellement avec l'attribut `AssertionConsumerServiceIndex` et il est normalement accompagné par l'attribut `ProtocolBinding`.

– `ProtocolBinding` [Facultatif]

Référence d'URI qui identifie une liaison de protocole SAML à utiliser lors du retour du message `<Response>`. Voir au paragraphe 10 des informations complémentaires sur les liaisons de protocole et les références d'URI définies pour elles. Cet attribut s'exclut mutuellement avec l'attribut `AssertionConsumerServiceIndex` et il est normalement accompagné par l'attribut `AssertionConsumerServiceURL`.

– `AttributeConsumingServiceIndex` [Facultatif]

Identifie indirectement les informations associées au demandeur qui décrivent les attributs SAML que le demandeur désire ou exige comme devant être fournies par le fournisseur d'identité dans le message `<Response>`. Le fournisseur d'identité doit avoir un moyen de confiance pour transposer la valeur d'indice dans l'attribut dans les informations associées au demandeur. Le paragraphe 9 donne un mécanisme possible. Le fournisseur d'identité peut utiliser ces informations pour remplir un ou plusieurs éléments `<saml:AttributeStatement>` dans la ou les assertions qu'il retourne.

– `ProviderName` [Facultatif]

Spécifie le nom sous forme lisible par l'homme du demandeur à utiliser par l'agent d'utilisateur du présentateur ou par le fournisseur d'identité.

Voir au paragraphe 8.2.4.4 les règles générales de traitement concernant ce message.

Le fragment de schéma suivant définit l'élément `<AuthnRequest>` et son type complexe **AuthnRequestType**:

```
<element name="AuthnRequest" type="samlp:AuthnRequestType"/>
<complexType name="AuthnRequestType">
      <complexContent>
            <extension base="samlp:RequestAbstractType">
                  <sequence>
                        <element ref="saml:Subject" minOccurs="0"/>
                        <element ref="samlp:NameIDPolicy" minOccurs="0"/>
                        <element ref="saml:Conditions" minOccurs="0"/>
                        <element ref="samlp:RequestedAuthnContext"
minOccurs="0"/>
                        <element ref="samlp:Scoping" minOccurs="0"/>
                  </sequence>
                  <attribute name="ForceAuthn" type="boolean"
use="optional"/>
                  <attribute name="IsPassive" type="boolean"
use="optional"/>
                  <attribute name="ProtocolBinding" type="anyURI"
use="optional"/>
```

```
                    <attribute name="AssertionConsumerServiceIndex"
type="unsignedShort" use="optional"/>
                    <attribute name="AssertionConsumerServiceURL"
type="anyURI" use="optional"/>
                    <attribute name="AttributeConsumingServiceIndex"
type="unsignedShort" use="optional"/>
                    <attribute name="ProviderName" type="string"
use="optional"/>
               </extension>
      </complexContent>
</complexType>
```

8.2.4.1 Élément <NameIDPolicy>

L'élément `<NameIDPolicy>` ajuste l'identifiant de nom dans les sujets des assertions qui résultent d'un `<AuthnRequest>`. Son type complexe **NameIDPolicyType** définit les attributs suivants:

– `Format` [Facultatif]

Spécifie la référence d'URI correspondant à un format d'identifiant de nom défini dans la présente Recommandation ou une autre (voir des exemples au § 8.7.3). Les valeurs supplémentaires de `urn:oasis:names:tc:SAML:2.0:nameid-format:encrypted` sont définies spécifiquement pour être utilisées au sein de cet attribut pour indiquer à une demande que l'identifiant résultant soit chiffré.

– `SPNameQualifier` [Facultatif]

Spécifie facultativement que l'identifiant du sujet d'assertion soit retourné (ou créé) dans l'espace de nom d'un fournisseur de service autre que le demandeur, ou dans l'espace de nom d'un groupe d'affiliation de fournisseurs de service. Voir par exemple la définition de `urn:oasis:names:tc:SAML:2.0:nameid-format:persistent` dans la présente Recommandation.

– `AllowCreate` [Facultatif]

Valeur booléenne utilisée pour indiquer si le fournisseur d'identité est admis, pendant le cours de la satisfaction de la demande, à créer un nouvel identifiant pour représenter le principal. La valeur par défaut est à "false". Lorsque la valeur est "false", le demandeur contraint le fournisseur d'identité à lui produire seulement une assertion si un identifiant acceptable pour le principal a déjà été établi. Cela n'empêche pas le fournisseur d'identité de créer de tels identifiants en dehors du contexte de cette demande spécifique (par exemple, à l'avance pour un grand nombre de principaux).

NOTE 1 (informative) – PE14 (voir OASIS PE:2006) apporte des éclaircissement à la définition ci-dessus comme suit:

une valeur booléenne utilisée pour indiquer si le demandeur accorde au fournisseur d'identité, dans le cours de la satisfaction de la demande, la permission de créer un nouvel identifiant ou d'associer un identifiant existant représentant le principal au consommateur d'assertions. La valeur par défaut est "false" s'il n'est pas présent ou si l'élément entier est omis.

NOTE 2 (informative) – PE14 (voir OASIS PE:2006) suggère d'ajouter le texte suivant au paragraphe ci-dessus:

l'attribut `AllowCreate` peut être utilisé par certains développements pour influencer la création d'états maintenus par le fournisseur d'identité se rapportant à l'utilisation d'un identifiant de nom (ou tout autre attribut persistant, identifiant de façon univoque) par un consommateur d'assertions particulier, pour des besoins tels que la création dynamique d'identifiant ou d'attribut, la recherche de consentement, l'utilisation ultérieure du protocole de gestion d'identifiant de nom, ou d'autres objets en rapport.

Lorsque la valeur est "false", le demandeur essaye de contraindre le fournisseur d'identité à ne produire une assertion que si un tel état a déjà été établi ou s'il juge applicable par le fournisseur d'identité d'utiliser un identifiant. Et donc, cela n'empêche pas le fournisseur d'identité de supposer que de telles informations existent en dehors du contexte de cette demande spécifique (par exemple, en l'établissant à l'avance pour un grand nombre de principaux).

Une valeur de "true" permet au fournisseur d'identité d'entreprendre toute action en rapport qu'il souhaite pour satisfaire la demande, sous réserve de toutes autres contraintes imposées par la demande et la politique (l'attribut `IsPassive`, par exemple).

Normalement, les demandeurs ne peuvent pas supposer des comportements spécifiques de la part des fournisseurs d'identité en ce qui concerne la création initiale ou l'association d'identifiants en leur nom, car ce sont des détails laissés à l'inititative des implémentations ou des développements. Absent des profils spécifiques qui gouvernent l'utilisation de cet attribut, il peut être utilisé comme conseil aux fournisseurs d'identité sur l'intention du demandeur de mémoriser l'identifiant ou de le lier à une valeur locale.

Une valeur de "false" pourrait être utilisée pour indiquer que le demandeur n'est pas prêt ou capable de le faire et épargner au fournisseur d'identité des efforts inutiles.

Les demandeurs qui n'ont pas d'utilisation spécifique de cet attribut devraient généralement le mettre à "true" pour maximiser l'interopérabilité. L'attribut `AllowCreate` ne doit pas être utilisé et devrait être ignoré dans les demandes ou les assertions produites avec des identifiants de nom avec un `Format` de

urn:oasis:names:tc:SAML:2.0:nameid-format:transient (ils empêchent un tel état dans et à partir d'eux).

Lorsque cet élément est utilisé, si le contenu n'est pas compris ou pas acceptable par le fournisseur d'identité, un élément de message <Response> doit être retourné avec un <Status> d'erreur, et peut contenir un <StatusCode> de second niveau de urn:oasis:names:tc:SAML:2.0:status:InvalidNameIDPolicy.

Si la valeur de Format est omise ou mise à urn:oasis:names:tc:SAML:2.0:nameid-format:unspecified, le fournisseur d'identité est alors libre de retourner toute sorte d'identifiant, sous réserve de toute contrainte supplémentaire due au contenu de cet élément ou des politiques du fournisseur d'identité ou principal.

La valeur Format urn:oasis:names:tc:SAML:2.0:nameid-format:encrypted spéciale indique que la ou les assertions résultantes doivent contenir des éléments <EncryptedID> au lieu de texte en clair. La forme non chiffrée de l'identifiant de nom sous-jacente peut être de tout type pris en charge par le fournisseur d'identité pour le sujet demandé.

NOTE 3 (informative) – PE6 (voir OASIS PE:2006) suggère d'ajouter le texte suivant à la fin de l'alinéa ci-dessus:

Il n'est pas possible au fournisseur de service de demander spécifiquement qu'une sorte d'identifiant particulière soit retournée s'il demande le chiffrement. L'élément de métadonnées <md:NameIDFormat> du § 9 ou d'autres moyens hors bande peuvent être utilisés pour déterminer quelle sorte d'identifiant chiffrer et retourner.

NOTE 4 (informative) – PE15 (voir OASIS PE:2006) suggère d'ajouter l'alinéa suivant:

Lorsqu'un Format défini au § 8.7.3.7 est utilisé, autre que urn:oasis:names:TC:SAML:2.0:nameid-format:unspecified ou urn:oasis:names:TC:SAML:2.0:nameid-format:encrypted, si le fournisseur d'identité retourne une assertion, alors:

- la valeur de Format du <NameID> au sein du <Subject> de toute <Assertion> doit être identique à la valeur de Format fournie dans le <NameIDPolicy>;
- si SPNameQualifier n'est pas omis dans <NameIDPolicy>, la valeur de SPNameQualifier du <NameID> au sein du <Subject> de toute <Assertion> doit être identique à la valeur de SPNameQualifier fournie dans le <NameIDPolicy>.

Quel que soit le Format dans le <NameIDPolicy>, le fournisseur d'identité peut retourner un <EncryptedID> dans le sujet d'assertion résultant si la politique en vigueur chez le fournisseur d'identité (qui peut être spécifique du fournisseur de service) exige d'utiliser un identifiant chiffré.

Si le demandeur souhaite permettre au fournisseur d'identité d'établir un nouvel identifiant pour le principal au cas où il n'en existe aucun, il doit inclure cet élément avec l'attribut AllowCreate mis à "true". Autrement, seul un principal pour lequel le fournisseur d'identité a précédemment établi un identifiant utilisable par le demandeur peut être authentifié avec succès. Ceci est principalement utile en conjonction avec la valeur urn:oasis:names:tc:SAML:2.0:nameid-format:persistent Format, (voir le § 12).

NOTE 5 (informative) – PE14 (voir OASIS PE:2006) suggère d'ignorer l'alinéa ci-dessus.

Le fragment de schéma suivant définit l'élément <NameIDPolicy> et son type complexe **NameIDPolicyType**:

```
<element name="NameIDPolicy" type="samlp:NameIDPolicyType"/>
<complexType name="NameIDPolicyType">
      <attribute name="Format" type="anyURI" use="optional"/>
      <attribute name="SPNameQualifier" type="string" use="optional"/>
      <attribute name="AllowCreate" type="boolean" use="optional"/>
</complexType>
```

8.2.4.2 Élément <Scoping>

L'élément <Scoping> spécifie les fournisseurs d'identité auxquels le demandeur fait confiance pour authentifier le présentateur, ainsi que les limitations et le contexte qui se rapportent au mandatement du message <AuthnRequest> aux fournisseurs d'identité suivants par le répondant. Son type complexe **ScopingType** définit les éléments et attributs suivants:

– ProxyCount [Facultatif]

Spécifie le nombre d'adressages indirects mandatés permis entre le fournisseur d'identité qui reçoit ce <AuthnRequest> et le fournisseur d'identité qui authentifie en dernier le principal. Un compte de zéro ne permet aucun mandatement, alors que l'omission de cet attribut exprime l'absence de restriction.

– <IDPList> [Facultatif]

Liste informative des fournisseurs d'identité et informations associées que le demandeur juge acceptables pour répondre à la demande.

– `<RequesterID>` [Zéro, un ou plusieurs]

Identifie l'ensemble des entités demandeuses au nom desquelles agit le demandeur. Utilisé pour communiquer la chaîne des demandeurs quand il y a du mandatement, comme décrit au paragraphe 8.2.4.5. Voir au paragraphe 8.7.3.6 une description des identifiants d'entité.

Dans les profils qui spécifient un intermédiaire actif, l'intermédiaire peut examiner la liste et retourner un message `<Response>` avec une erreur `<Status>` et un `<StatusCode>` de second niveau de `urn:oasis:names:tc:SAML:2.0:status:NoAvailableIDP` ou `urn:oasis:names:tc:SAML:2.0:status:NoSupportedIDP` si il ne peut contacter, ou ne veut prendre en charge, aucun des fournisseurs d'identité spécifiés.

Le fragment de schéma suivant définit l'élément `<Scoping>` et son type complexe **ScopingType**:

```
<element name="Scoping" type="samlp:ScopingType"/>
<complexType name="ScopingType">
      <sequence>
            <element ref="samlp:IDPList" minOccurs="0"/>
            <element ref="samlp:RequesterID" minOccurs="0"
maxOccurs="unbounded"/>
      </sequence>
      <attribute name="ProxyCount" type="nonNegativeInteger"
use="optional"/>
</complexType>
<element name="RequesterID" type="anyURI"/>
```

8.2.4.3 Élément <IDPList>

L'élément `<IDPList>` spécifie les fournisseurs d'identité en lesquels le demandeur a confiance pour authentifier le présentateur. Son type complexe **IDPListType** définit les éléments suivants:

– `<IDPEntry>` [Un ou plusieurs]

Informations sur un seul fournisseur d'identité.

– `<GetComplete>` [Facultatif]

Si la `<IDPList>` n'est pas complète, l'utilisation de cet élément spécifie une référence d'URI qui peut être utilisée pour récupérer la liste complète. La récupération des ressources associées à l'URI doit résulter en une instance XML dont l'élément racine est une `<IDPList>` qui ne contient pas elle-même d'élément `<GetComplete>`.

Le fragment de schéma suivant définit l'élément `<IDPList>` et son type complexe **IDPListType**:

```
<element name="IDPList" type="samlp:IDPListType"/>
<complexType name="IDPListType">
      <sequence>
            <element ref="samlp:IDPEntry" maxOccurs="unbounded"/>
            <element ref="samlp:GetComplete" minOccurs="0"/>
      </sequence>
</complexType>
<element name="GetComplete" type="anyURI"/>
```

L'élément `<IDPEntry>` spécifie un seul fournisseur d'identité, de confiance pour le demandeur, pour authentifier le présentateur. Son type complexe **IDPEntryType** définit les attributs suivants:

– `ProviderID` [Exigé]

Identifiant unique du fournisseur d'identité. Voir au paragraphe 8.7.3.6 une description de tels identifiants.

– `Name` [Facultatif]

Nom, lisible par l'homme, du fournisseur d'identité.

– `Loc` [Facultatif]

Référence d'URI qui représente la localisation d'un point d'extrémité spécifique d'un profil qui prend en charge le protocole de demande d'authentification. La liaison à utiliser doit être comprise du profil d'utilisation.

Le fragment de schéma suivant définit l'élément `<IDPEntry>` et son type complexe **IDPEntryType**:

```
<element name="IDPEntry" type="samlp:IDPEntryType"/>
<complexType name="IDPEntryType">
```

```
      <attribute name="ProviderID" type="anyURI" use="required"/>
      <attribute name="Name" type="string" use="optional"/>
      <attribute name="Loc" type="anyURI" use="optional"/>
</complexType>
```

8.2.4.4 Règles de traitement

L'échange <AuthnRequest> et <Response> prend en charge divers scénarios d'utilisation et est donc normalement profilé pour être utilisé dans un contexte spécifique dans lequel cette faculté est réfrénée et où des types spécifiques d'entrée et de sortie sont exigés ou prohibés. Les règles de traitement suivantes s'appliquent comme comportement invariant à travers tout profil de cet échange de protocole. Toutes les autres règles de traitement associées aux messages sous-jacents de demande et de réponse doivent aussi être observées.

Le répondant doit répondre en dernière instance à une <AuthnRequest> par un message <Response> contenant une ou plusieurs assertions qui satisfont aux spécifications définies par la demande, ou par un message <Response> contenant un <Status> décrivant les erreurs survenues. Le répondant peut avoir des échanges de messages supplémentaires avec le présentateur pour initialiser ou achever, autant que de besoin, le processus d'authentification, sous réserve de la nature de la liaison de protocole et du mécanisme d'authentification. Comme indiqué au paragraphe suivant, cela inclut de mandater la demande en dirigeant le présentateur sur un autre fournisseur d'identité en produisant son propre message <AuthnRequest>, de sorte que l'assertion qui en résulte puisse être utilisée pour authentifier le présentateur auprès du répondant d'origine, en utilisant en fait SAML comme mécanisme d'authentification.

Si le répondant n'est pas capable d'authentifier le présentateur ou ne reconnaît pas le sujet demandé, ou s'il est empêché de fournir une assertion par les politiques en vigueur chez le fournisseur d'identité (par exemple, le sujet prévu a interdit au fournisseur d'identité de fournir des assertions au consommateur d'assertions), il doit alors retourner une <Response> avec une erreur <Status>, et peut retourner un <StatusCode> de second niveau de: urn:oasis:names:tc:SAML:2.0:status:AuthnFailed; ou urn:oasis:names:tc:SAML:2.0:status:UnknownPrincipal.

Si l'élément <saml:Subject> est présent dans la demande, le <saml:Subject> des assertions résultantes doit **correspondre fortement** à la demande <saml:Subject>, comme décrit au § 8.2.3.4, excepté que l'identifiant peut être dans un format différent s'il en est spécifié ainsi par <NameIDPolicy>. Dans un tel cas, le contenu physique de l'identifiant peut être différent, mais il doit se référer au même principal.

Tous les contenus définis spécifiquement au sein de <AuthnRequest> sont facultatifs, bien que certains puissent être exigés par certains profils. L'absence de tout contenu spécifique implique le comportement suivant:

- la ou les assertions retournées doivent contenir un élément <saml:Subject> qui représente le présentateur. Les type et format d'identifiant sont déterminés par le fournisseur d'identité. Au moins une déclaration dans au moins une assertion doit être un <saml:AuthnStatement> qui décrit l'authentification effectuée par le répondant ou le service d'authentification qui lui est associé;
- le présentateur de la demande devrait, dans la mesure du possible, être la seule entité témoin capable de satisfaire le <saml:SubjectConfirmation> de la ou des assertions. Dans le cas de méthodes de confirmation plus faibles, on utilisera des mécanismes spécifiques de la liaison ou autres pour aider à satisfaire à cette exigence;
- la ou les assertions qui en résultent doivent contenir un élément <saml:AudienceRestriction> faisant référence au demandeur en tant que consommateur d'assertions acceptable. D'autres audiences peuvent être incluses selon que le fournisseur d'identité le juge approprié.

8.2.4.5 Mandatement

Si un fournisseur d'identité qui reçoit un <AuthnRequest> n'a pas encore authentifié le présentateur ou ne peut pas directement authentifier le présentateur, mais croit que le présentateur s'est déjà authentifié à un autre fournisseur d'identité ou un équivalent non-SAML, il peut répondre à la demande en produisant un nouveau <AuthnRequest> en son nom propre pour être présenté à l'autre fournisseur d'identité, ou une demande dans un format non-SAML que reconnaît l'entité. Le fournisseur d'identité original est dénommé fournisseur d'identité mandant.

Au retour réussi d'une <Response> (ou de son équivalent non-SAML) au fournisseur mandant, l'assertion qu'elle contient ou son équivalent non-SAML peut être utilisée pour authentifier le présentateur de telle sorte que le fournisseur mandant puisse produire une assertion de son cru en réponse à la <AuthnRequest> d'origine, terminant l'ensemble de l'échange de messages. Les fournisseurs d'identité mandant et authentificateur peuvent tous deux inclure des contraintes à l'activité de mandatement dans les messages et assertions qu'ils produisent, comme décrit aux paragraphes précédents et ci-dessous.

Le demandeur peut influencer le comportement du mandataire en incluant un élément <Scoping> où le fournisseur établit une valeur de ProxyCount désirée et/ou indique une liste de fournisseurs d'identité préférés qui peuvent être mandatés en incluant une <IDPList> ordonnée des fournisseurs préférés.

Un fournisseur d'identité peut contrôler l'utilisation secondaire de ses assertions par les fournisseurs d'identité mandatés en utilisant un élément <ProxyRestriction> dans les assertions qu'il produit.

Un fournisseur d'identité peut mandater un <AuthnRequest> si l'attribut <ProxyCount> est omis ou est supérieur à zéro. Le choix de mandater ou non est une affaire de politique locale. Un fournisseur d'identité peut choisir de mandater pour un fournisseur spécifié dans la liste <IDPList>, si elle est fournie, mais il n'est pas obligé de le faire.

Un fournisseur d'identité ne doit pas mandater une demande où <ProxyCount> est mis à zéro. Le fournisseur d'identité doit retourner une erreur <Status> contenant une valeur de <StatusCode> de second niveau de urn:oasis:names:tc:SAML:2.0:status:ProxyCountExceeded, sauf s'il peut authentifier directement le présentateur.

Si il choisit de mandater un fournisseur d'identité SAML, lorsqu'il crée le nouveau <AuthnRequest>, le fournisseur d'identité mandant doit inclure des formes équivalentes ou plus strictes de toutes les informations incluses dans la demande d'origine (comme la politique de contexte d'authentification). Cependant, le fait que le fournisseur mandant soit libre de spécifier le <NameIDPolicy> qu'il souhaite maximise les chances d'une réponse réussie.

Si le fournisseur d'identité authentificateur n'est pas un fournisseur d'identité SAML, le fournisseur mandant doit alors avoir un autre moyen de s'assurer que les éléments qui gouvernent l'interaction d'agent d'utilisateur (<IsPassive>, par exemple) seront honorés par le fournisseur authentificateur.

Le nouvel <AuthnRequest> doit contenir un attribut <ProxyCount> avec une valeur au plus inférieure de un à la valeur originale. Si la demande d'origine ne contient pas d'attribut <ProxyCount>, la nouvelle demande devrait alors contenir un attribut <ProxyCount>.

Si une <IDPList> été spécifiée dans la demande d'origine, la nouvelle demande doit aussi contenir une <IDPList>. Le fournisseur d'identité mandant peut ajouter des fournisseurs d'identité supplémentaires à la fin de <IDPList>, mais ne doit en retirer aucun de la liste.

La demande et réponse d'authentification est traitée de la façon normale, conformément aux règles données dans le présent paragraphe et au profil d'utilisation. Une fois que le présentateur s'est authentifié auprès du fournisseur d'identité mandant (dans le cas de SAML en délivrant une <Response>), on suit les étapes ci-après:

- le fournisseur d'identité mandant prépare une nouvelle assertion en son nom propre en copiant dans les informations pertinentes à partir de l'assertion d'origine ou son équivalent non-SAML;
- le <saml:Subject> de la nouvelle assertion doit contenir un identifiant qui satisfasse aux préférences du demandeur d'origine, comme défini par son élément <NameIDPolicy>;
- le <saml:AuthnStatement> dans la nouvelle assertion doit inclure un élément <saml:AuthnContext> contenant un élément <saml:AuthenticatingAuthority> référençant le fournisseur d'identité auquel le fournisseur d'identité mandant renvoie le présentateur. Si l'assertion d'origine contient des informations <saml:AuthnContext> qui incluent un ou plusieurs éléments <saml:AuthenticatingAuthority>, ces éléments devraient être inclus dans la nouvelle assertion, avec le nouvel élément placé après elles;
- si le fournisseur d'identité d'authentification n'est pas un fournisseur SAML, le fournisseur d'identité mandant doit générer une valeur d'identifiant univoque pour le fournisseur d'authentication. Cette valeur devrait être cohérente dans la durée pour les différentes demandes. La valeur ne doit pas entrer en conflit avec les valeurs utilisées ou générées par d'autres fournisseurs SAML;
- toutes les autres informations <saml:AuthnContext> peuvent être copiées, traduites, ou omises en conformité avec les politiques des fournisseurs d'identité mandants, pourvu que soient satisfaites les exigences originales dictées par le demandeur.

Si, à l'avenir, le fournisseur d'identité est appelé à authentifier le même présentateur pour un second demandeur, et si cette demande est également ou moins stricte que la demande d'origine (comme déterminé par le fournisseur d'identité mandant), le fournisseur d'identité peut sauter la création d'un nouvel <AuthnRequest> auprès du fournisseur d'identité authentifiant et produire immédiatement une autre assertion (en supposant que l'assertion originale ou son équivalent non-SAML qu'il a reçue est toujours valide).

8.2.5 Protocole de résolution d'artifice

Le protocole de résolution d'artifice fournit un mécanisme par lequel les messages de protocole SAML peuvent être transportés dans une liaison SAML par référence plutôt que par valeur. Les demandes et les réponses peuvent toutes

deux être obtenues par référence en utilisant ce protocole spécialisé. L'expéditeur d'un message, au lieu de lier un message à un protocole de transport, envoie un petit bout de données appelées un artifice en utilisant la liaison. Un artifice peut prendre diverses formes, mais doit accepter un moyen par lequel le receveur puisse déterminer qui est l'envoyeur. Si le receveur le souhaite, il peut alors utiliser ce protocole en conjonction avec un protocole de liaison SAML différent (généralement synchrone) pour résoudre l'artifice dans le message de protocole original.

L'utilisation la plus courante de ce mécanisme est en présence de liaisons qui ne peuvent pas transporter facilement un message à cause de contraintes de taille, ou pour permettre à un message d'être communiqué via un canal sécurisé entre le demandeur et le répondant SAML, en évitant le besoin de signature.

Selon les caractéristiques du message sous-jacent passé par référence, le protocole de résolution d'artifice exige des protections telles que l'authentification mutuelle, la protection de l'intégrité, la confidentialité, etc., de la part de la liaison de protocole utilisée pour résoudre l'artifice. Dans tous les cas, l'artifice doit exhiber une sémantique à usage unique telle qu'une fois qu'il a été résolu avec succès, elle ne puisse plus être utilisée par l'une ou l'autre partie.

Indépendamment du message de protocole obtenu, le résultat de la résolution d'un artifice doit être traité exactement comme si le message ainsi obtenu avait été envoyé à l'origine à la place de l'artifice.

8.2.5.1 Élément <ArtifactResolve>

Le message `<ArtifactResolve>` est utilisé pour demander qu'un message de protocole SAML soit retourné dans un message `<ArtifactResponse>` en spécifiant un artifice représentant le message de protocole SAML. La transmission originale de l'artifice est gouvernée par la liaison de protocole spécifique qui est utilisée.

Le message `<ArtifactResolve>` devrait être signé ou autrement authentifié et protégé en intégrité par la liaison de protocole utilisée pour délivrer le message.

Ce message a le type complexe **ArtifactResolveType**, qui étend **RequestAbstractType** et ajoute l'élément suivant:

– `<Artifact>` [Exigé]

  Valeur d'artifice que reçoit le demandeur et qu'il souhaite maintenant traduire dans le message de protocole qu'il représente.

Le fragment de schéma suivant définit l'élément `<ArtifactResolve>` et son type complexe **ArtifactResolveType**:

```
<element name="ArtifactResolve" type="samlp:ArtifactResolveType"/>
<complexType name="ArtifactResolveType">
       <complexContent>
              <extension base="samlp:RequestAbstractType">
                     <sequence>
                            <element ref="samlp:Artifact"/>
                     </sequence>
              </extension>
       </complexContent>
</complexType>
<element name="Artifact" type="string"/>
```

8.2.5.2 Élément <ArtifactResponse>

Le receveur d'un message `<ArtifactResolve>` doit répondre par un élément de message `<ArtifactResponse>`. Cet élément est du type complexe **ArtifactResponseType**, qui étend **StatusResponseType** avec un seul élément générique facultatif correspondant au message de protocole SAML retourné. Cet élément de message enveloppé peut être une demande ou une réponse.

Le message `<ArtifactResponse>` devrait être signé ou autrement authentifié et protégé en intégrité par la liaison de protocole utilisée pour délivrer le message.

Le fragment de schéma suivant définit l'élément `<ArtifactResponse>` et son type complexe **ArtifactResponseType**:

```
<element name="ArtifactResponse" type="samlp:ArtifactResponseType"/>
<complexType name="ArtifactResponseType">
       <complexContent>
              <extension base="samlp:StatusResponseType">
                     <sequence>
                            <any namespace="##any" processContents="lax"
minOccurs="0"/>
                     </sequence>
              </extension>
       </complexContent>
</complexType>
```

8.2.5.3 Règles de traitement

Si le répondant reconnaît l'artifice comme valide, il répond alors par le message de protocole associé dans un élément de message `<ArtifactResponse>`. Autrement, il répond par un élément `<ArtifactResponse>` sans message enchâssé. Dans les deux cas, l'élément `<Status>` doit inclure un élément `<StatusCode>` avec une valeur de code `urn:oasis:names:tc:SAML:2.0:status:Success`. Un message de réponse sans message enchâssé à l'intérieur est appelé une réponse vide dans la suite du présent paragraphe.

Le répondant doit mettre en application une propriété utilisable une seule fois sur l'artifice en s'assurant que toute réponse ultérieure avec le même artifice par tout demandeur résultera en une réponse vide, comme décrit ci-dessus.

Certains messages de protocole SAML, et plus particulièrement le message `<AuthnRequest>` dans certains profils, peuvent être destinés à être consommés par toute partie qui le reçoit et peut répondre de façon appropriée. Dans la plupart des autres cas, cependant, un message est destiné à une entité spécifique. Dans de tels cas, lorsqu'il est produit, l'artifice doit être associé au receveur prévu du message que l'artifice représente. Si le producteur de l'artifice reçoit un message `<ArtifactResolve>` d'un demandeur qui ne peut pas s'authentifier comme le receveur original prévu, le producteur de l'artifice doit alors retourner une réponse vide.

Le producteur de l'artifice devrait mettre en application la plus courte limite de temps pratique sur la durée d'utilisation d'un artifice, de telle sorte qu'une fenêtre temporelle acceptable (mais pas plus) existe pour que le receveur de l'artifice obtienne l'artifice et le retourne dans un message `<ArtifactResolve>` au producteur.

L'attribut `InResponseTo` du message `<ArtifactResponse>` doit contenir la valeur de l'attribut d'identifiant du message `<ArtifactResolve>` correspondant, mais le message de protocole enchassé contiendra son propre identifiant de message, et dans le cas d'une réponse enchassée, peut contenir une valeur de `InResponseTo` différente qui correspond au message de demande d'origine auquel le message enchassé répond.

Toutes les autres règles de traitement associées aux messages sous-jacents de demande et de réponse doivent être observées.

8.2.6 Protocole de gestion d'identifiant de nom

Après l'établissement d'un identifiant de nom pour un principal, un fournisseur d'identité qui souhaite changer la valeur et/ou le format de l'identifiant qu'il utilisera lorsqu'il se réfère au principal, ou pour indiquer qu'un identifiant de nom ne sera plus utilisé comme référence au principal, informe les fournisseurs de service du changement en leur envoyant un message `<ManageNameIDRequest>`.

NOTE 1 (informative) – PE12 (voir OASIS PE:2006) identifie les intentions de l'alinéa ci-dessus en le réécrivant comme suit:

> Après l'établissement d'un identifiant de nom pour un principal, un fournisseur d'identité qui souhaite changer la valeur de l'identifiant qu'il utilisera pour se référer au principal ou pour indiquer qu'un identifiant de nom ne sera plus utilisé comme référence au principal, informe les fournisseurs de service du changement en leur envoyant un message `<ManageNameIDRequest>`.

Un fournisseur de service utilise aussi ce message pour enregistrer ou changer la valeur `SPProvidedID` pour qu'elle soit incluse lorsque l'identifiant de nom sous-jacent est utilisé pour communiquer avec lui, ou pour mettre fin à l'utilisation d'un identifiant de nom entre lui-même et le fournisseur d'identité.

Ce protocole n'est normalement pas utilisé avec des identifiants de nom "transitoires", car leur valeur n'est pas destinée à être gérée sur une base à long terme.

NOTE 2 (informative) – PE14 (see OASIS PE:2006) précise le texte ci-dessus comme suit:

> Ce protocole ne doit pas être utilisé en conjonction avec le `Format` de `<NameID>` `urn:oasis:names:tc:SAML:2.0:nameidformat:transient`.

8.2.6.1 Élément <ManageNameIDRequest>

Un fournisseur envoie un message `<ManageNameIDRequest>` pour informer le receveur d'un changement d'identifiant de nom ou pour indiquer la fin d'utilisation d'un identifiant de nom.

Le message `<ManageNameIDRequest>` devrait être signé ou autrement authentifié et protégé en intégrité par la liaison de protocole utilisée pour délivrer le message.

Ce message a le type complexe **ManageNameIDRequestType**, qui étend **RequestAbstractType** et ajoute les éléments suivants:

– `<saml:NameID>` ou `<saml:EncryptedID>` [Exigé]

  Identifiant de nom et données descriptives associées (en texte clair ou sous forme chiffrée) qui spécifie le principal comme normalement reconnu par l'identité et le fournisseur de services avant cette demande (pour des informations complémentaires sur cet élément, voir au § 8.1.2).

– <NewID> ou <NewEncryptedID> ou <Terminate> [Exigé]

Nouvelle valeur d'identifiant (en clair ou sous forme chiffrée) à utiliser pour communiquer avec le fournisseur demandeur concernant ce principal, ou pour indiquer que l'utilisation du vieil identifiant est terminée. Dans le premier cas, si le demandeur est le fournisseur de service, le nouvel identifiant doit apparaître dans les éléments <NameID> suivants dans l'attribut SPProvidedID. Si le demandeur est le fournisseur d'identité, la nouvelle valeur apparaîtra dans les éléments <NameID> suivants comme contenu de l'élément.

NOTE (informative) – PE12 (voir OASIS PE:2006) suggère d'ajouter ce qui suit au paragraphe ci-dessus:

Dans tous les cas, si le <NewEncryptedID> est utilisé, son contenu chiffré est juste un élément <NewID> qui ne contient que la nouvelle valeur pour l'identifiant (format et qualifiants ne peuvent être changés une fois établis).

Le fragment de schéma suivant définit l'élément <ManageNameIDRequest> et son type complexe **ManageNameIDRequestType**:

```
<element name="ManageNameIDRequest" type="samlp:ManageNameIDRequestType"/>
<complexType name="ManageNameIDRequestType">
      <complexContent>
            <extension base="samlp:RequestAbstractType">
                  <sequence>
                        <choice>
                              <element ref="saml:NameID"/>
                              <element ref="saml:EncryptedID"/>
                        </choice>
                        <choice>
                              <element ref="samlp:NewID"/>
                              <element ref="samlp:NewEncryptedID"/>
                              <element ref="samlp:Terminate"/>
                        </choice>
                  </sequence>
            </extension>
      </complexContent>
</complexType>
<element name="NewID" type="string"/>
<element name="NewEncryptedID" type="saml:EncryptedElementType"/>
<element name="Terminate" type="samlp:TerminateType"/>
<complexType name="TerminateType"/>
```

8.2.6.2 Élément <ManageNameIDResponse>

Le receveur d'un message <ManageNameIDRequest> doit répondre par un message <ManageNameIDResponse>, qui est du type **StatusResponseType** sans contenu supplémentaire.

Le message <ManageNameIDResponse> devrait être signé ou autrement authentifié et protégé en intégrité par la liaison de protocole utilisée pour délivrer le message.

Le fragment de schéma suivant définit l'élément <ManageNameIDResponse>:

```
<element name="ManageNameIDResponse" type="samlp:StatusResponseType"/>
```

8.2.6.3 Règles de traitement

Si la demande inclut un <saml:NameID> (ou sa version chiffrée) que le receveur ne reconnaît pas, le fournisseur répondant doit répondre par une erreur <Status> et peut répondre par un <StatusCode> de second niveau de urn:oasis:names:tc:SAML:2.0:status:UnknownPrincipal.

NOTE 1 (informative) – PE14 (voir OASIS PE:2006) précise l'alinéa ci-dessous. Se reporter à l'Appendice VIII pour les détails.

Si l'élément <Terminate> est inclus dans la demande, le fournisseur demandeur indique que (dans le cas d'un fournisseur de service) il n'acceptera plus d'assertions de ce fournisseur d'identité ou (dans le cas d'un fournisseur d'identité) il ne produira plus d'assertions à ce fournisseur de service sur le principal. Le fournisseur receveur peut effectuer toute opération de maintenance en ayant la connaissance que la relation représentée par l'identifiant de nom a été terminée. Il peut choisir d'invalider la ou les sessions actives pour un principal pour lequel les relations sont terminées.

NOTE 2 (informative) – PE8 (voir OASIS PE:2006) suggère de remplacer la dernière phrase du présent alinéa par:

En général il ne devrait pas invalider de session active du principal pour lequel les relations sont terminées. Si le fournisseur receveur est un fournisseur d'identité, il ne devrait pas invalider de session active du principal établie avec d'autres fournisseurs de service. Un fournisseur demandeur peut envoyer un message <LogoutRequest> avant d'initialiser une terminaison d'identifiant de nom en envoyant un message <ManageNameIDRequest> si c'est l'intention du fournisseur demandeur (par exemple, la terminaison d'identifiant de nom est initialisée via un administrateur qui souhaite terminer toutes

les activités d'utilisateur). Le fournisseur demandeur ne doit pas envoyer un message <LogoutRequest> après l'envoi du message <ManageNameIDRequest>.

Si le fournisseur de service demande que son identifiant pour le principal soit changé en y incluant un élément <NewID> (ou <NewEncryptedID>), le fournisseur d'identité doit inclure le contenu de l'élément comme SPProvidedID lorsqu'il communique ultérieurement avec le fournisseur de service à propos de ce principal.

Si le fournisseur d'identité demande que son identifiant pour le principal soit changé en incluant un élément <NewID> (ou <NewEncryptedID>), le fournisseur de service doit utiliser le contenu de l'élément comme contenu d'élément <saml:NameID> lorsqu'il communique ultérieurement avec le fournisseur d'identité à propos de ce principal.

L'identifiant original et le nouveau peuvent être chiffrés tous les deux, l'un ou l'autre ou ni l'un ni l'autre (en utilisant les éléments <EncryptedID> et <NewEncryptedID>).

Dans tous les cas, le contenu de <saml:NameID> dans la demande et son attribut SPProvidedID associé doivent contenir les informations les plus récentes d'identifiant de nom établies entre les fournisseurs pour le principal.

Dans le cas d'un identifiant avec un Format de urn:oasis:names:tc:SAML:2.0:nameid-format:persistent, l'attribut NameQualifier doit contenir l'identifiant univoque du fournisseur d'identité qui a créé l'identifiant. Si l'identifiant a été établi entre le fournisseur d'identité et un groupe d'affiliation dont le fournisseur de service est membre, l'attribut SPNameQualifier doit contenir l'identifiant univoque du groupe d'affiliation. Autrement, il doit contenir l'identifiant univoque du fournisseur de service. Ces attributs peuvent être omis si ils correspondraient autrement à la valeur de l'élément <Issuer> du message de protocole contenant, mais ce n'est pas recommandé, comme opportunité de confusion.

Le changement de ces identifiants peut prendre un temps pouvant être significatif pour se propager à travers les systèmes, à la fois du demandeur et du répondant. Les implémentations pourraient souhaiter permettre à chaque partie d'accepter les deux identifiants pendant un certain délai après l'achèvement réussi d'un changement d'identifiant de nom. Ne pas le faire pourrait avoir pour résultat l'incapacité du principal à accéder aux ressources.

Toutes les autres règles de traitement associées aux messages sous-jacents de demande et de réponse doivent être observées.

8.2.7 Protocole de fermeture de session unique

Le protocole de fermeture de session unique fournit un protocole d'échange de messages par lequel toutes les sessions fournies par une autorité de session particulière sont terminées presque simultanément. Le protocole de fermeture de session unique est utilisé lorsqu'un principal ferme la session à un participant de session ou quand un principal ferme la session directement à l'autorité de session. Ce protocole peut aussi être utilisé pour fermer la session à un principal du fait d'une fin de temporisation. La raison de l'événement de fermeture de session peut être indiqué au moyen de l'attribut Reason.

Le principal peut avoir établi des sessions authentifiées avec l'autorité de session et avec les participants individuels à la session, sur la base d'assertions contenant des déclarations d'authentification fournies par l'autorité de session.

Lorsque le principal invoque le processus de fermeture de session unique auprès d'un participant de session, celui-ci doit envoyer un message <LogoutRequest> à l'autorité de session qui a fourni au participant de session l'assertion contenant la déclaration d'authentification qui se rapporte à cette session.

Lorsque le principal invoque une fermeture de session auprès de l'autorité de session, ou qu'un participant de session envoie une demande de fermeture de session à l'autorité de session en spécifiant ce principal, l'autorité de session devrait envoyer un message <LogoutRequest> à chaque participant de session auquel il a fourni des assertions contenant des déclarations d'authentification dans sa session en cours avec le principal, à l'exception du participant de session qui a envoyé le message <LogoutRequest> à l'autorité de session. Il devrait essayer de contacter autant de ces participants qu'il peut en utilisant ce protocole, terminer sa propre session avec le principal, et finalement retourner un message <LogoutResponse> au participant de session demandeur, s'il en est.

8.2.7.1 Élément <LogoutRequest>

Un participant de session ou une autorité de session envoie un message <LogoutRequest> pour indiquer qu'une session s'est terminée.

Le message <LogoutRequest> devrait être signé ou autrement authentifié et protégé en intégrité par la liaison de protocole utilisée pour délivrer le message.

Ce message a le type complexe **LogoutRequestType**, qui étend **RequestAbstractType** et ajoute les éléments et attributs suivants:

– `NotOnOrAfter` [Facultatif]

Moment auquel la demande arrive à expiration, après lequel le receveur peut éliminer le message. La valeur horaire est codée en UTC, comme décrit au § 7.3.

– `Reason` [Facultatif]

Indication de la raison de la fermeture de la session, sous forme d'une référence d'URI.

NOTE 1 (informative) – PE10 (voir OASIS PE:2006) suggère de remplacer le texte ci-dessus par:

L'attribut `Reason` est spécifié comme une chaîne dans le schéma. La présente spécification apporte des restrictions supplémentaires au schéma en exigeant que l'attribut `Reason` soit de la forme référence d'URI.

– `<saml:BaseID>` ou `<saml:NameID>` ou `<saml:EncryptedID>` [Exigé]

Identifiant et attributs associés (en clair ou sous forme chiffrée) qui spécifient le principal tel que couramment reconnu par l'identité et le fournisseur de services avant cette demande. (Pour des informations complémentaires sur cet élément, voir au paragraphe 8.1.2.)

– `<SessionIndex>` [Facultatif]

Identifiant qui indexe cette session auprès du receveur du message.

NOTE 2 (informative) – PE38 (voir OASIS PE:2006) précise le texte ci-dessus comme suit:

Indice de la session entre le principal identifié par l'élément `<saml:BaseID>`, `<saml:NameID>`, ou `<saml:EncryptedID>`, et l'autorité de session. Cela doit se corréler à l'attribut `SessionIndex`, s'il en est, dans le `<saml:AuthnStatement>` de l'assertion utilisée pour établir la session qui est en train de se terminer."

Le fragment de schéma suivant définit l'élément `<LogoutRequest>` et le type complexe **LogoutRequestType** associé:

```
<element name="LogoutRequest" type="samlp:LogoutRequestType"/>
    <complexType name="LogoutRequestType">
        <complexContent>
            <extension base="samlp:RequestAbstractType">
                <sequence>
                    <choice>
                        <element ref="saml:BaseID"/>
                        <element ref="saml:NameID"/>
                        <element ref="saml:EncryptedID"/>
                    </choice>
                    <element ref="samlp:SessionIndex" minOccurs="0"
maxOccurs="unbounded"/>
                </sequence>
                <attribute name="Reason" type="string" use="optional"/>
                <attribute name="NotOnOrAfter" type="dateTime"
use="optional"/>
            </extension>
        </complexContent>
    </complexType>
    <element name="SessionIndex" type="string"/>
```

8.2.7.2 Élément <LogoutResponse>

Le receveur d'un message `<LogoutRequest>` doit répondre par un message `<LogoutResponse>`, de type **StatusResponseType**, sans contenu supplémentaire spécifié.

Le message `<LogoutResponse>` devrait être signé ou autrement authentifié et protégé en intégrité par la liaison de protocole utilisée pour délivrer le message.

Le fragment de schéma suivant définit l'élément `<LogoutResponse>`:

```
<element name="LogoutResponse" type="samlp:StatusResponseType"/>
```

8.2.7.3 Règles de traitement

L'envoyeur du message peut utiliser l'attribut `Reason` pour indiquer la raison de l'envoi de `<LogoutRequest>`. Les valeurs suivantes sont définies par la présente Recommandation pour utilisation par tous les expéditeurs de message; d'autres valeurs peuvent être agréées entre participants:

`urn:oasis:names:tc:SAML:2.0:logout:user`

Spécifie que le message est envoyé parce que le principal souhaite terminer la session indiquée.

`urn:oasis:names:tc:SAML:2.0:logout:admin`

Spécifie que le message est envoyé parce qu'un administrateur souhaite terminer la session indiquée pour ce principal.

Toutes les autres règles de traitement associées aux messages sous-jacents de demande et de réponse doivent être observées.

Des règles de traitement supplémentaires sont fournies dans les paragraphes suivants.

1) Règle de participant de session

Lorsqu'un participant de session reçoit un message `<LogoutRequest>`, le participant de session doit authentifier le message. Si l'envoyeur est l'autorité qui a fourni une assertion contenant une déclaration d'authentification liée à la session en cours du principal, le participant de session doit invalider la ou les sessions du principal qui sont visées par l'élément `<saml:BaseID>`, `<saml:NameID>`, ou `<saml:EncryptedID>`, et tous éléments `<SessionIndex>` fournis dans le message. Si aucun élément `<SessionIndex>` n'est fourni, toutes les sessions associées au principal doivent alors être invalidées.

Le participant de session doit appliquer le message de demande de fermeture de session à toute assertion qui satisfait aux conditions suivantes, même si l'assertion arrive après la demande de fermeture de session:

- le sujet de l'assertion **correspond fortement** à l'élément `<saml:BaseID>`, `<saml:NameID>`, ou `<saml:EncryptedID>` dans le `<LogoutRequest>`, comme défini au paragraphe 8.2.3.4;
- l'attribut `SessionIndex` d'une des déclarations d'authentification de l'assertion correspond à un des éléments `<SessionIndex>` spécifiés dans la demande de fermeture de session, ou la demande de fermeture de session ne contient pas d'élément `<SessionIndex>`;
- l'assertion serait autrement valide, sur la base des conditions d'horaire spécifiées dans l'assertion elle-même (en particulier, la valeur de tout attribut `NotOnOrAfter` spécifié dans les données de confirmation de condition ou de sujet);

la demande de fermeture de session n'a pas encore expiré (déterminé en examinant l'attribut `NotOnOrAfter` du message).

NOTE – Cette règle est destinée à empêcher une situation dans laquelle un participant de session reçoit une demande de fermeture de session ciblée sur un seule, ou plusieurs assertions (comme identifié par le ou les éléments `<SessionIndex>`) *avant* qu'il ne reçoive la ou les assertions réelles – et qui peuvent être toujours valides – visées par la demande de fermeture de session. Il devrait honorer la demande de fermeture de session jusqu'à ce que la demande de fermeture de session elle-même puisse être éliminée (la valeur `NotOnOrAfter` sur la demande a été dépassée) ou l'assertion visée par la demande de fermeture de session a été reçue et traitée de façon appropriée.

2) Règles d'autorité de session

Lorsqu'une autorité de session reçoit un message `<LogoutRequest>`, l'autorité de session doit authentifier l'envoyeur. Si l'envoyeur est un participant de session auquel l'autorité de session a fourni une assertion contenant une déclaration d'authentification pour la session en cours, l'autorité de session devrait alors faire ce qui suit dans l'ordre spécifié:

– envoyer un message `<LogoutRequest>` à toute autorité de session au nom de laquelle l'autorité de session a mandaté l'authentification du principal , sauf si la seconde autorité est l'origine de la `<LogoutRequest>`;

– envoyer un message `<LogoutRequest>` à chaque participant de session pour lequel l'autorité de session a fourni des assertions dans la session en cours, *autre que* l'origine d'une `<LogoutRequest>` en cours;

– terminer la session en cours du principal comme spécifié par l'élément `<saml:BaseID>`, `<saml:NameID>`, ou `<saml:EncryptedID>`, et tous éléments `<SessionIndex>` présents dans le message de demande de fermeture de session.

Si l'autorité de session termine avec succès la session du principal par rapport à elle-même, elle doit alors répondre au demandeur d'origine, s'il en est, avec un message `<LogoutResponse>` contenant un code d'état de niveau supérieur de `urn:oasis:names:tc:SAML:2.0:status:Success`. S'il ne peut le faire, il doit alors répondre par un message `<LogoutResponse>` contenant un code d'état de niveau supérieur indiquant l'erreur. Et donc, l'état de niveau supérieur indique l'état de l'opération de fermeture de session uniquement par rapport à l'autorité de session elle-même.

L'autorité de session devrait essayer de contacter chaque participant de session en utilisant toute liaison applicable/utilisable de protocole, même si un ou plusieurs de ces essais échoue ou ne peut pas être tenté (par exemple parce que la demande d'origine a lieu en utilisant une liaison de protocole qui ne permet pas à la fermeture de session d'être propagée à tous les participants).

Dans le cas où tous les participants de session ne répondent pas avec succès à ces messages `<LogoutRequest>` (ou si tous les participants ne peuvent être contactés), l'autorité de session doit alors inclure dans son message `<LogoutResponse>` un code d'état de second niveau de `urn:oasis:names:tc:SAML:2.0:status:PartialLogout` pour indiquer que tous les autres participants de session n'ont pas réussi à répondre avec la confirmation de la terminaison de session.

Une autorité de session peut initialiser une terminaison de session pour des raisons autres que celle d'avoir reçu une `<LogoutRequest>` d'un participant de session – cela inclut, sans s'y limiter:

- si une période de temporisation avait été acceptée hors bande d'un commun accord avec un participant de session individuel, l'autorité de session peut envoyer une `<LogoutRequest>` à ce seul participant individuel;
- une période globale de temporisation acceptée d'un commun accord a été dépassée;
- le principal ou une autre entité de confiance a demandé la fermeture de session du principal directement à l'autorité de session;
- l'autorité de session a déterminé que les accréditifs du principal pourraient avoir été compromis.

Lors de la construction d'un message de demande de fermeture de session, l'autorité de session doit régler la valeur de l'attribut `NotOnOrAfter` du message à une valeur de temps, indiquant une heure d'expiration pour le message, après laquelle la demande de fermeture de session peut être éliminée par le receveur. Cette valeur devrait être réglée à une valeur de temps égale ou supérieure à la valeur de tout attribut `NotOnOrAfter` spécifié dans l'assertion la plus récemment produite au titre de la session visée (comme indiqué par l'attribut `SessionIndex` sur la demande de fermeture de session).

En plus des valeurs spécifiées au § 8.2.6.3 pour l'attribut `Reason`, les valeurs suivantes sont aussi disponibles pour l'utilisation par la seule autorité de session:

`urn:oasis:names:tc:SAML:2.0:logout:global-timeout`

Spécifie que le message est envoyé parce que la période d'intervalle de temporisation global de session est dépassée.

`urn:oasis:names:tc:SAML:2.0:logout:sp-timeout`

Spécifie que le message est envoyé parce qu'une période d'intervalle de temporisation accepté d'un commun accord entre un participant et l'autorité de session a été dépassée.

8.2.8 Protocole de mappage d'identifiant de nom

Lorsqu'une entité qui partage un identifiant pour un principal avec un fournisseur d'identité souhaite obtenir un identifiant de nom pour le même principal dans un format ou espace de nom de fédération particulier, il peut envoyer une demande au fournisseur d'identité en utilisant ce protocole.

Par exemple, un fournisseur de service, qui souhaite communiquer avec un autre fournisseur de service avec lequel il ne partage pas d'identifiant pour le principal, peut utiliser un fournisseur d'identité qui partage un identifiant pour le principal avec les deux fournisseurs de service pour mapper son propre identifiant en un nouvel identifiant, généralement chiffré, avec lequel il peut communiquer avec le second fournisseur de service.

Indépendamment du type d'identifiant impliqué, l'identifiant mappé devrait être chiffré en un élément `<saml:EncryptedID>` sauf si un développement spécifique indique qu'une telle protection n'est pas nécessaire.

8.2.8.1 Élément <NameIDMappingRequest>

Pour demander un identifiant de nom de remplacement pour un principal de la part d'un fournisseur d'identité, un demandeur envoie un message `<NameIDMappingRequest>`. Ce message a le type complexe **NameIDMappingRequestType**, qui étend **RequestAbstractType** et ajoute les éléments suivants:

– `<saml:BaseID>` ou `<saml:NameID>` ou `<saml:EncryptedID>` [Exigé]

Identifiant et données descriptives associées qui spécifient que le principal est actuellement reconnu par le demandeur et le répondant. (Pour plus d'informations sur cet élément, voir au § 8.1.2.)

– `<NameIDPolicy>` [Exigé]

Exigences concernant le format et le qualificatif de nom facultatif pour l'identifiant à retourner.

Le message devrait être signé ou autrement authentifié et protégé en intégrité par la liaison de protocole utilisée pour délivrer le message.

Le fragment de schéma suivant définit l'élément <NameIDMappingRequest> et son type complexe **NameIDMappingRequestType**:

```
<element name="NameIDMappingRequest"
type="samlp:NameIDMappingRequestType"/>
<complexType name="NameIDMappingRequestType">
       <complexContent>
              <extension base="samlp:RequestAbstractType">
                     <sequence>
                            <choice>
                                   <element ref="saml:BaseID"/>
                                   <element ref="saml:NameID"/>
                                   <element ref="saml:EncryptedID"/>
                            </choice>
                            <element ref="samlp:NameIDPolicy"/>
                     </sequence>
              </extension>
       </complexContent>
</complexType>
```

8.2.8.2 Élément <NameIDMappingResponse>

Le receveur d'un message <NameIDMappingRequest> doit répondre par un message <NameIDMappingResponse>. Ce message a le type complexe **NameIDMappingResponseType**, qui étend **StatusResponseType** et ajoute les éléments suivants:

– <saml:NameID> ou <saml:EncryptedID> [Exigé]

Identifiant et attributs associés qui spécifient le principal de la façon requise, habituellement en forme chiffrée. (Pour plus d'informations sur cet élément, voir au § 8.1.2.)

Le message devrait être signé ou autrement authentifié et protégé en intégrité par la liaison de protocole utilisée pour délivrer le message.

Le fragment de schéma suivant définit l'élément <NameIDMappingResponse> et son type complexe **NameIDMappingResponseType**:

```
<element name="NameIDMappingResponse"
type="samlp:NameIDMappingResponseType"/>
<complexType name="NameIDMappingResponseType">
       <complexContent>
              <extension base="samlp:StatusResponseType">
                     <choice>
                            <element ref="saml:NameID"/>
                            <element ref="saml:EncryptedID"/>
                     </choice>
              </extension>
       </complexContent>
</complexType>
```

8.2.8.3 Règles de traitement

Si le répondant ne reconnaît pas le principal identifié dans la demande, il peut répondre par une erreur <Status> contenant un <StatusCode> de second niveau de:

urn:oasis:names:tc:SAML:2.0:status:UnknownPrincipal.

A la discrétion du répondant, le code d'état urn:oasis:names:tc:SAML:2.0:status:InvalidNameIDPolicy peut être retourné pour indiquer l'incapacité à fournir ou la volonté de ne pas fournir un identifiant dans le format ou espace de nom demandé.

Toutes les autres règles de traitement associées aux messages sous-jacents de demande et de réponse doit être observées.

8.3 Versions de SAML

L'ensemble des Recommandation SAML est muni d'un numéro de version de deux façons indépendantes. Chacune est exposée dans les paragraphes suivants, avec les règles de traitement pour détecter et traiter les différences de version. Sont aussi incluses des lignes directrices sur quand et pourquoi des changements des informations spécifiques de la version sont attendues dans les révisions futures de SAML.

Quand des informations de version sont exprimées à la fois comme version majeure et version mineure, c'est exprimé sous la forme *Major.Minor*. Le numéro de version $Major_B.Minor_B$ est plus élevé que le numéro de version $Major_A.Minor_A$ si et seulement si:

$$(Major_B > Major_{A)} \text{ OU } (\ (\ Major_B = Major_A\ ) \text{ ET } (Minor_B > Minor_A\ ))$$

8.3.1 Version d'ensemble de spécification SAML

Chaque livraison de la Recommandation SAML contiendra une désignation de version majeure et mineure décrivant ses relations avec les versions antérieures et ultérieures de la Recommandation. La version sera exprimée dans le contenu de la Recommandation. La taille globale et la portée des changements à la Recommandation indiqueront de façon informelle si un ensemble de changements constitue une révision majeure ou mineure. En général, si les changements cumulés sont rétro-compatibles avec une version antérieure, la nouvelle version sera une révision mineure. Autrement, les changements constitueront une révision majeure.

La présente Recommandation définit la version V2.0.

8.3.1.1 Version de schéma

Comme mécanisme de documentation non-normatif, tout document de schéma XML publié au titre de l'ensemble de spécifications contiendra un attribut version sur l'élément `<xs:schema>` dont la valeur est sous la forme *Major.Minor*, reflétant la version de l'ensemble de spécifications dans laquelle elle a été publiée. Les implémentations valides peuvent utiliser cet attribut comme moyen de distinguer quelle version d'un schéma est utilisée pour valider les messages, ou pour prendre en charge plusieurs versions du même schéma logique.

8.3.1.2 Version d'assertion SAML

L'élément SAML `<Assertion>` contient un attribut pour exprimer la version majeure et mineure de l'assertion dans une chaîne de la forme *Major.Minor*. Chaque version de l'ensemble de spécifications SAML sera construit de sorte à documenter la syntaxe, la sémantique, et les règles de traitement des assertions de la même version. C'est-à-dire que l'ensemble de specifications de version 1.0 décrit les assertions de version 1.0, et ainsi de suite.

Il n'y a explicitement PAS de relation entre la version d'assertion et l'espace de nom XML cible spécifié pour les définitions de schéma pour cette version d'assertion.

Les règles de traitement suivantes s'appliquent:

- un producteur d'assertions SAML ne doit pas produire d'assertion avec un numéro de version d'assertion global *Major.Minor* non pris en charge par l'autorité;
- un consommateur d'assertions SAML ne doit pas traiter d'assertion avec un numéro de version d'assertion majeur non pris en charge par le consommateur d'assertions;
- un consommateur d'assertions SAML peut traiter ou peut rejeter une assertion dont le numéro de version d'assertion mineur est supérieur au numéro de version d'assertion mineur pris en charge par le consommateur d'assertions. Cependant, toutes les assertions qui partagent un numéro de version d'assertion majeur doivent partager les mêmes règles générales de traitement et de sémantique, et peuvent être traitées d'une façon uniforme par une implémentation. Par exemple, si une assertion de V1.1 partage la syntaxe d'une assertion V1.0, une implémentation peut traiter l'assertion comme une assertion V1.0 sans effet négatif.

8.3.1.3 Version du protocole SAML

Les divers éléments de demande et de réponse du protocole SAML contiennent un attribut pour exprimer la version majeure et mineure de la demande ou message de réponse en utilisant une chaîne de forme *Major.Minor*. Chaque version de l'ensemble de spécifications SAML sera construit de façon à documenter la syntaxe, la sémantique, et les règles de traitement des messages de protocole de même version. C'est-à-dire que l'ensemble de spécifications de version 1.0 décrit les demandes et réponse de version V1.0, et ainsi de suite.

Il n'y a explicitement PAS de relation entre la version de protocole et l'espace de nom XML cible spécifié pour les définitions de schéma pour cette version de protocole.

Les numéros de version utilisés dans les éléments de demande et réponse du protocole SAML correspondront pour toute révision particulière de l'ensemble de spécifications SAML.

1) Version de demande

Les règles de traitement suivantes s'appliquent à la demande:

- un demandeur SAML devrait produire une demande avec le plus haut numéro de version de demande pris en charge à la fois par le demandeur SAML et le répondant SAML;
- si le demandeur SAML ne connaît pas les capacités du répondant SAML, il devrait alors supposer que le répondant accepte les demandes avec la version de demande la plus élevée que prend en charge le demandeur;
- un demandeur SAML ne doit pas produire de message de demande avec un numéro de version de demande *Major.Minor* globak correspondant à un numéro de version de réponse que le demandeur n'accepte pas;
- un répondant SAML doit rejeter toute demande avec un numéro de version de demande majeur non pris en charge par le répondant.

Un répondant SAML peut traiter ou peut rejeter toute demande dont le numéro de version de demande mineur est supérieur à la plus haute version de demande acceptée qu'il prend en charge. Cependant, toute demande qui partage un numéro de version de demande majeur doit partager les mêmes règles générales de traitement et de sémantique, et peut être traitée de façon uniforme par une implémentation. C'est-à-dire que, si une demande V1.1 partage la syntaxe d'une demande V1.0, un répondant peut traiter le message de demande comme une demande V1.0 sans effet négatif.

2) Version de réponse

Les règles de traitement suivantes s'appliquent aux réponses:

- un répondant SAML ne doit pas produire de message de réponse avec un numéro de version de réponse supérieur au numéro de version de la demande du message de demande correspondant;
- un répondant SAML ne doit pas produire de message de réponse avec un numéro de version de réponse majeur inférieur au numéro de version majeure de la demande du message de demande correspondant sauf pour rapporter l'erreur `urn:oasis:names:tc:SAML:2.0:status:RequestVersionTooHigh`;
- une réponse d'erreur résultant de versions incompatibles du protocole SAML doit avoir pour résultat le rapport d'une valeur `<StatusCode>` de niveau supérieur `urn:oasis:names:tc:SAML:2.0:status:VersionMismatch`, et peut résulter en le rapport d'une des valeurs de second niveau suivantes:
  - `urn:oasis:names:tc:SAML:2.0:status:RequestVersionTooHigh`;
  - `urn:oasis:names:tc:SAML:2.0:status:RequestVersionTooLow`;
  - `urn:oasis:names:tc:SAML:2.0:status:RequestVersionDeprecated`.

3) Combinaisons de version permises

Les assertions d'une version majeure particulière n'apparaissent qu'en réponse à des messages de la même version majeure, comme l'autorise l'importation d'espace de nom d'assertion SAML dans le schéma de protocole SAML. Par exemple, une assertion V1.1 peut apparaître dans un message de réponse V1.0, et une assertion V1.0 dans un message de réponse V1.1, si le schéma d'assertion approprié est référencé durant l'importation d'espace de nom. Mais une assertion V1.0 ne doit pas apparaître dans un message de réponse V2.0 parce qu'elles sont de versions majeures différentes.

8.3.2 Version d'espace de nom SAML

Les documents de schéma XML publiés au titre de cet ensemble de spécifications contiennent un ou plusieurs espaces de nom cibles dans lesquels sont placées les définitions de type, d'élément, et d'attribut. Chaque espace de nom est distinct des autres, et représente, en abrégé, les définitions structurelles et syntaxiques qui constituent cette partie de la spécification.

Les références d'URI d'espace de nom définies par l'ensemble de spécifications contiendront généralement des informations sur la version sous la forme *Major.Minor* quelque part dans l'URI. Les versions majeure et mineure dans l'URI doivent correspondre aux versions majeure et mineure de l'ensemble de spécifications dans lequel l'espace de nom est introduit et défini pour la première fois. Cette information n'est normalement pas consommée par un processeur XML, qui traite l'espace de nom de façon opaque, mais elle est destinée à communiquer les relations entre l'ensemble de spécifications et les espaces de nom qu'il définit. Ce schéma est aussi suivi par les identifiants fondés sur des URI définis par SAML dont la liste figure au § 8.7.

A titre de règle générale, les développeurs peuvent s'attendre à ce que les espaces de nom et les définitions de schéma associées définis par une révision majeure de l'ensemble de spécifications restent valides et stables à travers les révisions mineures de la spécification. De nouveaux espaces de nom peuvent être introduits, et lorsque nécessaire, de

vieux espaces de nom remplacés, mais on s'attend à ce que ce soit rare. Dans de tels cas, les espaces de nom les plus anciens et leur définition associée devraient être supposés rester valides jusqu'à une révision majeure de l'ensemble de spécifications.

En général, maintenir la stabilité des espaces de nom tout en ajoutant ou changeant le contenu d'un schéma sont des objectifs contradictoires. Alors que certaines stratégies de conception peuvent faciliter de tels changements, il est compliqué de prédire comment les plus anciennes implémentations vont réagir à un changement donné, ce qui rend la compatibilité aval difficile à réaliser. Quoi qu'il en soit, le droit de faire de tels changements dans les révisions mineures est suspendu, dans l'intérêt de la stabilité de l'espace de nom. Sauf circonstances particulières (par exemple, pour corriger des déficiences majeures ou réparer des erreurs), les implémentations devraient s'attendre à des modifications de schéma compatibles vers l'aval dans les révisions mineures, permettant aux nouveaux messages d'être valides dans les schémas plus anciens.

Les implémentations devraient s'attendre et être prêtes à traiter de nouvelles extensions et de nouveaux types de message conformément aux règles de traitement posées pour ces types. Les révisions mineures peuvent introduire de nouveaux types démultipliant les facilités d'extension décrites dans la présente Recommandation. Les implémentations plus anciennes devraient rejeter de telles extensions de bonne grâce lorsqu'elle les rencontrent dans des contextes qui imposent une sémantique obligatoire. A titre d'exemple, on citera de nouveaux types d'interrogation, de déclaration, ou de condition.

8.4 SAML et syntaxe et traitement de signature XML

Les assertions SAML et les messages de demande et réponse du protocole SAML peuvent être signés, avec les bénéfices suivants. Une assertion signée par le producteur d'assertions prend en charge l'intégrité de l'assertion, l'authentification du producteur d'assertions auprès d'un consommateur d'assertions SAML, et, si la signature est fondée sur la paire de clés publique/privée de l'autorité SAML, la non-répudiation d'origine. Un message de demande et de réponse de protocole SAML signé par le générateur du message prend en charge l'intégrité du message, l'authentification de l'origine du message à une destination, et, si la signature se fonde sur la paire de clés publique/privée du générateur, la non-répudiation de l'origine.

Une signature numérique n'est pas toujours nécessaire dans SAML. Par exemple, dans certaines circonstances, les signatures peuvent être "héritées," comme lorsque une assertion non signée obtient la protection d'une signature sur le message de réponse du protocole contenant. Les signatures "héritées" devraient être utilisées avec précaution lorsque l'objet contenu (comme l'assertion) est destiné à avoir une durée de vie non transitoire. La raison en est que c'est tout le contexte qui doit être examiné pour permettre la validation, exposant le contenu XML et ajoutant une redondance potentielle non nécessaire. Parmi d'autres exemples, le consommateur d'assertions SAML ou le demandeur SAML peut avoir obtenu une assertion ou message de protocole de la part du producteur d'assertions SAML ou du répondant SAML directement (sans intermédiaire) à travers un canal sécurisé, le producteur d'assertions ou répondant SAML s'étant authentifié auprès du consommateur d'assertions ou répondant SAML par un moyen autre qu'une signature numérique.

De nombreuses techniques différentes sont disponibles pour l'authentification "directe" et l'établissement de canaux sécurisés entre deux parties. La liste inclut TLS, HMAC, des mécanismes fondés sur un mot de passe, et ainsi de suite. De plus, les exigences de sécurité applicables dépendent des applications communiquantes et de la nature de l'assertion ou du message transporté. Il est recommandé que, dans tout autre contexte, les signatures numériques soient utilisées pour les assertions et messages de demande et réponse. Précisément:

- une assertion SAML obtenue par un consommateur d'assertions SAML d'une entité autre que le producteur d'assertions SAML devrait être signée par le producteur d'assertions SAML;
- un message de protocole SAML arrivant à destination en provenance d'une entité autre que l'expéditeur d'origine devrait être signée par l'envoyeur;
- les profils peuvent spécifier des mécanismes de signature de remplacement tels que S/MIME ou les objets Java signés qui contiennent des documents SAML. Les mises en garde sur la prise en compte du contexte et de l'interopérabilité s'appliquent. Les signatures XML sont destinées à être le principal mécanisme de signature SAML, mais la présente Recommandation essaye d'assurer la compatibilité avec les profils qui peuvent exiger d'autres mécanismes;
- sauf si un profil spécifie un mécanisme de signature de remplacement, toute signature numérique XML doit être enveloppée.

8.4.1 Signature des assertions

Toutes les assertions SAML peuvent être signées en utilisant la signature XML. Ceci est reflété dans le schéma d'assertion décrit au § 8.

8.4.2 Signature de demande/réponse

Tout message de demande et réponses de protocole SAML peut être signé en utilisant la signature XML. Ceci est reflété dans le schéma décrit à l'Annexe A.

8.4.3 Héritage de signature

Une assertion SAML peut être enchassée au sein d'un autre élément SAML, tel qu'une `<Assertion>` enveloppante ou une demande ou réponse, qui peut être signée. Lorsqu'une assertion SAML ne contient pas d'élément `<ds:Signature>`, mais est contenue dans un élément SAML enveloppant qui contient un élément `<ds:Signature>`, et que la signature s'applique à l'élément `<Assertion>` et à tous ses enfants, l'assertion peut alors être considérée comme héritière de la signature provenant de l'élément enveloppant. L'interprétation qui en résulte devrait être équivalente au cas où l'assertion elle-même était signée avec les mêmes options de clé et de signature.

De nombreux cas d'utilisation de SAML impliquent des données XML SAML enveloppées dans d'autres structures de données protégées telles que les messages SOAP signés, les paquetages S/MIME, et les connexions TLS authentifiées. Les profils SAML peuvent définir des règles supplémentaires pour interpréter les éléments SAML comme héritiers de signatures ou autres informations d'authentification de la part du contexte environnant, mais un tel héritage ne devrait être supposé que s'il est spécifiquement identifié par le profil.

8.4.4 Profil de signature XML

XML Signature:2002 du W3C fournit une syntaxe XML générale souple et comportant de nombreux choix pour les données de signature. Le présent paragraphe précise les contraintes qui pèsent sur ces facilités de sorte que les processeurs SAML n'aient pas à traiter de tout le processus général des signatures XML. Ce traitement fait une utilisation spécifique des attributs **xs:ID**-typed présents sur les éléments racine auquels les signatures peuvent s'appliquer, particulièrement l'attribut `ID` (d'identifiant) sur `<Assertion>` et les divers éléments de demande et réponse. Ces attributs sont collectivement désignés dans ce paragraphe comme les attributs d'identifiant.

Ce profil ne s'applique qu'à l'utilisation des éléments `<ds:Signature>` qui se trouvent directement au sein des assertions SAML, demandes, et réponses. Les autres profils dans lesquels apparaissent ailleurs des signatures mais qui s'appliquent au contenu SAML ont toute liberté pour définir d'autres approches.

8.4.4.1 Formats et algorithmes de signature

La signature XML a trois façons de rapporter une signature à un document: enveloppante, enveloppée, et détachée.

Les assertions et protocoles SAML doivent utiliser les signatures enveloppées lors de la signature des assertions et des messages de protocole. Les processeurs SAML devraient prendre en charge l'utilisation de signature et vérification RSA pour les opérations à clés publiques conformément à l'algorithme identifié au § 6.4 de http://www.w3.org/2000/09/xmldsig#rsa-sha1.

8.4.4.2 Références

Les assertions SAML et les messages de protocole doivent fournir une valeur pour l'attribut `ID` (d'identifiant) sur l'élément racine de l'assertion ou du message de protocole à signer. L'élément racine de l'assertion ou du message de protocole peut ou non être l'élément racine du document XML réel qui contient l'assertion ou message de protocole signé (par exemple, il pourrait être contenu dans une enveloppe SOAP).

Les signatures doivent contenir un seul `<ds:Reference>` contenant une référence same-document à la valeur d'attribut `ID` (d'identifiant) de l'élément racine de l'assertion ou message de protocole à signer. Par exemple, si la valeur d'attribut `ID` (d'identifiant) est "foo", l'attribut d'URI dans l'élément `<ds:Reference>` doit alors être "#foo".

8.4.4.3 Méthode de canonisation

Les implémentations SAML devraient utiliser la canonisation exclusive, avec ou sans commentaire, à la fois dans l'élément `<ds:CanonicalizationMethod>` de `<ds:SignedInfo>`, et comme algorithme `<ds:Transform>`. L'utilisation de la canonisation exclusive garantit que les signatures créées sur les messages SAML enchassés dans un contexte XML peuvent être vérifiées indépendamment de ce contexte.

8.4.4.4 Transformations

Les signatures dans les messages SAML ne devraient pas contenir de transformations autres que la transformation de signature enveloppée (avec l'identifiant http://www.w3.org/2000/09/xmldsig#enveloped-signature) ou la transformation de canonisation exclusive (avec l'identifiant http://www.w3.org/2001/10/xml-exc-c14n# ou http://www.w3.org/2001/10/xml-exc-c14n#WithComments).

Les vérificateurs de signatures peuvent rejeter comme invalides les signatures qui contiennent d'autres algorithmes de transformation. S'ils ne le font pas, ils doivent s'assurer qu'aucun contenu du message SAML n'est exclu de la signature.

Ceci peut être réalisé en établissant un accord hors bande comme quoi les transformations sont acceptables, ou en appliquant manuellement les transformations au contenu et en revérifiant pour voir si le résultat donne le même message SAML.

8.4.4.5 KeyInfo

La signature W3C définit l'usage de l'élément `<ds:KeyInfo>`. SAML n'exige pas l'usage de `<ds:KeyInfo>`, ni n'impose de restriction sur son usage. Donc, `<ds:KeyInfo>` peut être absent.

8.4.4.6 Exemple

Ci-après figure un exemple de réponse signée contenant une assertion signée. Les coupures de ligne ont été ajoutées pour faciliter la lecture; les signatures ne sont pas valides et ne peuvent être vérifiées avec succès.

```
<Response
  IssueInstant="2003-04-17T00:46:02Z" Version="2.0"
  ID="_c7055387-af61-4fce-8b98-e2927324b306"
  xmlns="urn:oasis:names:tc:SAML:2.0:protocol"
  xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion">
       <saml:Issuer>https://www.opensaml.org/IDP"</saml:Issuer>
       <ds:Signature xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
              <ds:SignedInfo>
                     <ds:CanonicalizationMethod
                            Algorithm="http://www.w3.org/2001/10/xml-exc-
c14n#"/>
                     <ds:SignatureMethod
                            Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-
sha1"/>
                     <ds:Reference URI="#_c7055387-af61-4fce-8b98-
e2927324b306">
                            <ds:Transforms>
                                   <ds:Transform

       Algorithm="http://www.w3.org/2000/09/xmldsig#enveloped-signature"/>
                                   <ds:Transform

       Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#">
                                          <InclusiveNamespaces
PrefixList="#default saml ds xs xsi"

       xmlns="http://www.w3.org/2001/10/xml-exc-c14n#"/>
                                   </ds:Transform>
                            </ds:Transforms>
                            <ds:DigestMethod

       Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>

       <ds:DigestValue>TCDVSuG6grhyHbzhQFWFzGrxIPE=</ds:DigestValue>
                     </ds:Reference>
              </ds:SignedInfo>
              <ds:SignatureValue>

       x/GyPbzmFEe85pGD3c1aXG4Vspb9V9jGCjwcRCKrtwPS6vdVNCcY5rHaFPYWkf+5

       EIYcPzx+pX1h43SmwviCqXRjRtMANWbHLhWAptaK1ywS7gFgsD01qjyen3CP+m3D
                     w6vKhaqledl0BYyrIzb4KkHO4ahNyBVXbJwqv5pUaE4=
              </ds:SignatureValue>
              <ds:KeyInfo>
                     <ds:X509Data>
                            <ds:X509Certificate>

       MIICyjCCAjOgAwIBAgICAnUwDQYJKoZIhvcNAQEEBQAwgakxCzAJBgNVBAYTAlVT

       MRIwEAYDVQQIEwlXaXNjb25zaW4xEDAOBgNVBAcTB01hZGlzb24xIDAeBgNVBAoT

       F1VuaXZlcnNpdHkgb2YgV2lzY29uc2luMSswKQYDVQQLEyJEaXZpc2lvbiBvZiBJ

       bmZvcm1hdGlvbiBUZWNobm9sb2d5MSUwIwYDVQQDExxIRVBLSSBTZXJ2ZXIgQ0Eg

       LS0gMjAwMjA3MDFBMB4XDTAyMDcyNjA3Mjc1MVoXDTA2MDkwNDA3Mjc1MVowgYsx
```

```
      CzAJBgNVBAYTAlVTMREwDwYDVQQIEwhNaWNoaWdhbjESMBAGA1UEBxMJQW5uIEFy

      Ym9yMQ4wDAYDVQQKEwVVQ0FJRDEcMBoGA1UEAxMTc2hpYjEuaW50ZXJuZXQyLmVk

      dTEnMCUGCSqGSIb3DQEJARYYcm9vdEBzaGliMS5pbnRlcm5ldDIuZWR1MIGfMA0G

      CSqGSIb3DQEBAQUAA4GNADCBiQKBgQDZSAb2sxvhAXnXVIVTx8vuRay+x50z7GJj

      IHRYQgIv6IqaGG04eTcyVMhoekE0b45QgvBIaOAPSZBl13R6+KYiE7x4XAWIrCP+

      c2MZVeXeTgV3Yz+USLg2Y1on+Jh4HxwkPFmZBctyXiUr6DxF8rvoP9W7O27rhRjE

      pmqOIfGTWQIDAQABox0wGzAMBgNVHRMBAf8EAjAAMAsGA1UdDwQEAwIFoDANBgkq

      hkiG9w0BAQQFAAOBgQBfDqEW+OI3jqBQHIBzhujN/PizdN7s/z4D5d3pptWDJf2n

      qgi7lFV6MDkhmTvTqBtjmNk3No7v/dnP6Hr7wHxvCCRwubnmIfZ6QZAv2FU78pLX

      8I3bsbmRAUg4UP9hH6ABVq4KQKMknxu1xQxLhpR1ylGPdiowMNTrEG8cCx3w/w==
                        </ds:X509Certificate>
                  </ds:X509Data>
            </ds:KeyInfo>
      </ds:Signature>
      <Status>
            <StatusCode
Value="urn:oasis:names:tc:SAML:2.0:status:Success"/>
      </Status>
      <Assertion ID="_a75adf55-01d7-40cc-929f-dbd8372ebdfc"
            IssueInstant="2003-04-17T00:46:02Z" Version="2.0"
            xmlns="urn:oasis:names:tc:SAML:2.0:assertion">
            <Issuer>https://www.opensaml.org/IDP</Issuer>
            <ds:Signature xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
                  <ds:SignedInfo>
                        <ds:CanonicalizationMethod
                              Algorithm="http://www.w3.org/2001/10/xml-
exc-c14n#"/>
                        <ds:SignatureMethod

      Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-sha1"/>
                        <ds:Reference URI="#_a75adf55-01d7-40cc-929f-
dbd8372ebdfc">
                              <ds:Transforms>
                                    <ds:Transform

      Algorithm="http://www.w3.org/2000/09/xmldsig#enveloped-signature"/>
                                    <ds:Transform

      Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#">
                                          <InclusiveNamespaces
                                                PrefixList="#default
saml ds xs xsi"

      xmlns="http://www.w3.org/2001/10/xml-exc-c14n#"/>
                                    </ds:Transform>
                              </ds:Transforms>
                              <ds:DigestMethod

      Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>

      <ds:DigestValue>Kclet6XcaOgOWXM4gty6/UNdviI=</ds:DigestValue>
                        </ds:Reference>
                  </ds:SignedInfo>
                  <ds:SignatureValue>

      hq4zk+ZknjggCQgZm7ea8fI79gJEsRy3E8LHDpYXWQIgZpkJN9CMLG8ENR4Nrw+n

      7iyzixBvKXX8P53BTCT4VghPBWhFYSt9tHWu/AtJfOTh6qaAsNdeCyG86jmtp3TD
                        MwuL/cBUj2OtBZOQMFn7jQ9YB7klIz3RqVL+wNmeWI4=
                  </ds:SignatureValue>
                  <ds:KeyInfo>
```

```
                        <ds:X509Data>
                              <ds:X509Certificate>

      MIICyjCCAjOgAwIBAgICAnUwDQYJKoZIhvcNAQEEBQAwgakxCzAJBgNVBAYTAlVT

      MRIwEAYDVQQIEwlXaXNjb25zaW4xEDAOBgNVBAcTB01hZGlzb24xIDAeBgNVBAoT

      F1VuaXZlcnNpdHkgb2YgV2lzY29uc2luMSswKQYDVQQLEyJEaXZpc2lvbiBvZiBJ

      bmZvcm1hdGlvbiBUZWNobm9sb2d5MSUwIwYDVQQDExxIRVBLSSBTZXJ2ZXIgQ0Eg

      LS0gMjAwMjA3MDFBMB4XDTAyMDcyNjA3Mjc1MVoXDTA2MDkwNDA3Mjc1MVowgYsx

      CzAJBgNVBAYTAlVTMREwDwYDVQQIEwhNaWNoaWdhbjESMBAGA1UEBxMJQW5uIEFy

      Ym9yMQ4wDAYDVQQKEwVVQ0FJRDEcMBoGA1UEAxMTc2hpYjEuaW50ZXJuZXQyLmVk

      dTEnMCUGCSqGSIb3DQEJARYYcm9vdEBzaGliMS5pbnRlcm5ldDIuZWR1MIGfMA0G

      CSqGSIb3DQEBAQUAA4GNADCBiQKBgQDZSAb2sxvhAXnXVIVTx8vuRay+x50z7GJj

      IHRYQgIv6IqaGG04eTcyVMhoekE0b45QgvBIaOAPSZBl13R6+KYiE7x4XAWIrCP+

      c2MZVeXeTgV3Yz+USLg2Y1on+Jh4HxwkPFmZBctyXiUr6DxF8rvoP9W7O27rhRjE

      pmqOIfGTWQIDAQABox0wGzAMBgNVHRMBAf8EAjAAMAsGA1UdDwQEAwIFoDANBgkq

      hkiG9w0BAQQFAAOBgQBfDqEW+OI3jqBQHIBzhujN/PizdN7s/z4D5d3pptWDJf2n

      qgi7lFV6MDkhmTvTqBtjmNk3No7v/dnP6Hr7wHxvCCRwubnmIfZ6QZAv2FU78pLX

      8I3bsbmRAUg4UP9hH6ABVq4KQKMknxu1xQxLhpR1ylGPdiowMNTrEG8cCx3w/w==
                              </ds:X509Certificate>
                        </ds:X509Data>
                  </ds:KeyInfo>
            </ds:Signature>
            <Subject>
                  <NameID
                        Format="urn:oasis:names:tc:SAML:1.1:nameid-
format:emailAddress">
                        scott@example.org
                  </NameID>
                  <SubjectConfirmation
                        Method="urn:oasis:names:tc:SAML:2.0:cm:bearer"/>
            </Subject>
            <Conditions NotBefore="2003-04-17T00:46:02Z"
                        NotOnOrAfter="2003-04-17T00:51:02Z">
                  <AudienceRestriction>
                        <Audience>http://www.opensaml.org/SP</Audience>
                  </AudienceRestriction>
            </Conditions>
            <AuthnStatement AuthnInstant="2003-04-17T00:46:00Z">
                  <AuthnContext>
                        <AuthnContextClassRef>

      urn:oasis:names:tc:SAML:2.0:ac:classes:Password
                        </AuthnContextClassRef>
                  </AuthnContext>
            </AuthnStatement>
      </Assertion>
</Response>
```

8.5 Syntaxe et traitement du chiffrement SAML et XML

Le chiffrement est utilisé comme moyen d'implémenter la confidentialité. Le motif le plus courant de la confidentialité est de protéger la vie privée des individus ou de protéger des secrets d'organisations pour un avantage concurrentiel ou des raisons similaires. La confidentialité peut aussi être nécessaire pour assurer l'efficacité de quelque autre mécanisme de sécurité. Par exemple, un mot de passe secret ou une clé peuvent être chiffrés.

Plusieurs moyens sont fournis pour utiliser le chiffrement pour la protection de confidentialité de tout ou partie d'une assertion SAML.

- La confidentialité des communications peut être fournie par des mécanismes associés à des liaisons ou profils particuliers. Par exemple, la liaison SOAP prend en charge l'utilisation de TLS (voir la RFC 2246 de l'IETF) ou les mécanismes de sécurité de messages SOAP pour la confidentialité.
- Un `<SubjectConfirmation>` secret peut être protégé par l'utilisation de l'élément `<ds:KeyInfo>` au sein de `<SubjectConfirmationData>`, qui permet aux clés ou autres secrets d'être chiffrés.
- Un élément `<Assertion>` entier peut être chiffré, comme décrit au § 8.1.3.4.
- L'élément `<BaseID>` ou `<NameID>` peut être chiffré, comme décrit au § 8.1.2.4.
- Un élément `<Attribute>` peut être chiffré, comme décrit au § 8.1.7.3.2.

8.5.1 Considérations générales

Le chiffrement des éléments `<Assertion>`, `<BaseID>`, `<NameID>` et `<Attribute>` est fourni par l'utilisation du chiffrement XML. Les données chiffrées et facultativement une ou plusieurs clés chiffrées doivent remplacer les informations de texte en clair dans la même localisation au sein de l'instance XML. L'attribut Type de l'élément `<EncryptedData>` devrait être utilisé et, s'il est présent, doit avoir la valeur http://www.w3.org/2001/04/xmlenc#Element.

NOTE (informative) – PE30 (voir OASIS PE:2006) suggère de remplacer un ou plusieurs à la seconde ligne par zéro, un ou plusieurs.

Tous les algorithmes définis pour être utilisés avec le Chiffrement XML peuvent être utilisés pour effectuer le chiffrement. Le schéma SAML est défini de telle sorte que l'inclusion des données chiffrée donne une instance valide.

8.5.2 Combinaison des signatures et du chiffrement

L'utilisation du chiffrement XML et de la signature XML peut être combinée. Quand une assertion doit être signée et chiffrée, les règles suivantes s'appliquent. Un consommateur d'assertions doit effectuer la validation de signature et le déchiffrement dans l'ordre inverse de celui où la signature et le chiffrement ont été effectués.

- Lorsqu'un élément `<Assertion>` signé est chiffré, la signature doit d'abord être calculée et placée dans l'élément `<Assertion>` avant que celui-ci soit chiffré.
- Lorsqu'un élément `<BaseID>`, `<NameID>`, ou `<Attribute>` est chiffré, le chiffrement doit être effectué d'abord et la signature calculée ensuite sur l'assertion ou message contenant l'élément chiffré.

8.6 Extensibilité de SAML

SAML prend en charge l'extensibilité d'un certain nombre de façons, qui incluent l'extension des schémas d'assertion et de protocole. Voir à la section Profils SAML dans la présente Recommandation des informations sur la façon de définir de nouveaux profils, qui peut être combinée avec des extensions pour mettre de nouvelles utilisations dans le cadre de travail SAML.

8.6.1 Extension de schéma

Les éléments dans les schémas SAML sont empêchés de se prêter à des substitutions, ce qui signifie qu'aucun élément SAML ne peut servir d'élément de tête d'un groupe de substitution. Cependant, la définition des types SAML n'est pas fermée, de sorte que tous les types SAML peuvent être étendus et restreints. En pratique, cela signifie que les extensions sont normalement définies seulement en tant que types plutôt qu'en tant qu'éléments, et sont incluses dans des instances SAML au moyen d'un attribut `xsi:type`.

Les paragraphes suivants exposent seulement les éléments et types qui ont été spécifiquement conçus pour prendre en charge l'extensibilité.

8.6.1.1 Extension de schéma d'assertion

Le schéma d'assertion SAML est conçu pour permettre un traitement séparé du paquetage assertion et des déclarations qu'il contient, si le mécanisme d'extension est utilisé pour chaque partie.

Les éléments suivants sont destinés spécifiquement à être utilisés comme points d'extension dans un schéma d'extension; leurs types sont réglés à abstract, et ils ne sont donc utilisables que comme base d'un type dérivé:

- `<BaseID>` et **BaseIDAbstractType**
- `<Condition>` et **ConditionAbstractType**
- `<Statement>` et **StatementAbstractType**

Les constructions suivantes qui sont directement utilisables au titre de SAML sont des cibles particulièrement intéressantes pour l'extension:

- `<AuthnStatement>` et **AuthnStatementType**
- `<AttributeStatement>` et **AttributeStatementType**
- `<AuthzDecisionStatement>` et **AuthzDecisionStatementType**
- `<AudienceRestriction>` et **AudienceRestrictionType**
- `<ProxyRestriction>` et **ProxyRestrictionType**
- `<OneTimeUse>` et **OneTimeUseType**

8.6.1.2 Extension de schéma de protocole

Les éléments de protocole SAML suivants sont destinés spécifiquement à une utilisation comme points d'extension dans un schéma d'extension; leurs types sont réglés à abstract, et ils ne sont donc utilisables que comme base d'un type dérivé:

- `<Request>` et **RequestAbstractType**
- `<SubjectQuery>` et **SubjectQueryAbstractType**

Les constructions suivantes qui sont directement utilisables au titre de SAML sont des cibles particulièrement intéressantes pour l'extension:

- `<AuthnQuery>` et **AuthnQueryType**
- `<AuthzDecisionQuery>` et **AuthzDecisionQueryType**
- `<AttributeQuery>` et **AttributeQueryType**
- **StatusResponseType**

8.6.2 Points d'extension génériques de schéma

Les schémas SAML utilisent des constructions génériques dans certaines localisations pour permettre d'utiliser des éléments et attributs provenant d'espaces de nom arbitraires, qui servent de point d'extension incorporé sans qu'il soit besoin d'un schéma d'extension.

8.6.2.1 Points d'extension d'assertion

Les constructions suivantes dans le schéma d'assertion permettent des constructions provenant d'espaces de nom arbitraires en leur sein:

- `<SubjectConfirmationData>`: utilise **xs:anyType**, qui admet tous sous-éléments et attributs.
- `<AuthnContextDecl>`: utilise **xs:anyType**, qui admet tous sous-éléments et attributs.
- `<AttributeValue>`: utilise **xs:anyType**, qui admet tous sous-éléments et attributs.
- `<Advice>` et **AdviceType**: en plus des éléments SAML natifs, permet des éléments provenant d'autres espaces de nom avec un processus souple de validation de schéma.

La construction suivante dans le schéma d'assertion admet des attributs globaux arbitraires:

- `<Attribute>` et **AttributeType**

8.6.2.2 Points d'extension de protocole

Les constructions suivantes dans le schéma de protocole permettent des constructions provenant d'espaces de nom arbitraires en leur sein:

- `<Extensions>` et **ExtensionsType**: permet des éléments provenant d'autres espaces de nom avec un processus souple de validation de schéma.
- `<StatusDetail>` et **StatusDetailType**: permet des éléments provenant d'autres espaces de nom avec un processus souple de validation de schéma.
- `<ArtifactResponse>` et **ArtifactResponseType**: permet des éléments provenant d'autres espaces de nom avec un processus souple de validation de schéma. (Il est cependant spécifiquement destiné à porter un élément de message de demande SAML ou réponse SAML.)

8.6.3 Extension d'identifiant

SAML utilise des identifiants fondés sur des URI pour un certain nombre d'objets, tels que les codes d'état et les formats d'identifiant de nom, et définit certains identifiants qui peuvent être utilisés à cet effet; la plupart figurent dans la liste du § 8.7. Cependant, il est toujours possible de définir des identifiants supplémentaires fondés sur des URI à cet effet. Il est recommandé que ces identifiants suplémentaires soient définis dans un profil d'utilisation formel. La signification d'un URI donné utilisé comme un tel identifiant ne devrait en aucun cas changer de façon significative, ou être utilisée pour désigner deux choses différentes.

8.7 Identifiants définis dans SAML

Les paragraphes qui suivent définissent des identifiants à base d'URI pour des actions d'accès aux ressources communes, des formats d'identifiant de sujet de nom, et des formats de nom d'attribut.

Lorsque c'est possible, on utilise un URN existant pour spécifier un protocole. Dans le cas des protocoles de l'IETF, on utilise l'URN de la RFC la plus courante qui spécifie ce protocole. Les références d'URI créées spécialement pour SAML ont une des souches suivantes, selon la version de l'ensemble de spécifications dans lequel elles ont d'abord été introduites:

```
urn:oasis:names:tc:SAML:1.0:
urn:oasis:names:tc:SAML:1.1:
urn:oasis:names:tc:SAML:2.0:
```

La présente Recommandation introduit la dernière souche.

8.7.1 Identifiants d'espace de nom d'action

Les identifiants suivants peuvent être utilisés dans l'attribut d'espace de nom de l'élément `<Action>` pour désigner des ensembles communs d'actions à effectuer sur des ressources.

8.7.1.1 Lire/Ecrire/Exécuter/Supprimer/Contrôler

URI: `urn:oasis:names:tc:SAML:1.0:action:rwedc`

Actions définies: `Read Write Execute Delete Control`

Ces actions s'interprètent comme suit:

`Read`: le sujet peut lire la ressource.

`Write`: le sujet peut modifier la ressource.

`Execute`: le sujet peut exécuter la ressource.

`Delete`: le sujet peut supprimer la ressource.

`Control`: le sujet peut spécifier la politique de contrôle d'accès pour la ressource.

8.7.1.2 Lire/Ecrire/Exécuter/Supprimer/Contrôler avec négation

URI: `urn:oasis:names:tc:SAML:1.0:action:rwedc-negation`

Actions définies: `Read Write Execute Delete Control ~Read ~Write ~Execute ~Delete ~Control`

Les actions spécifiées au § 8.7.1.1 sont interprétées de la même manière que décrit ici. Les actions préfixées avec un tilde (~) sont des permissions négatives et sont utilisées pour spécifier de façon affirmative que la permission déclarée est déniée. Et donc un sujet décrit comme étant autorisé à effectuer l'action ~Read se voit affirmativement dénier la permission de lire.

Une autorité SAML ne doit pas autoriser à la fois une action et sa forme négative.

8.7.1.3 Get/Head/Put/Post

URI: `urn:oasis:names:tc:SAML:1.0:action:ghpp`

Actions définies: `GET HEAD PUT POST`

Ces actions sont liées aux opérations HTTP correspondantes. Par exemple, un sujet autorisé à effectuer l'action `GET` sur une ressource est autorisée à l'obtenir.

Les actions `GET` et `HEAD` correspondent en gros à la permission conventionnelle de lecture et les actions `PUT` et `POST` à la permission d'écriture. La correspondance n'est cependant pas exacte car l'opération GET de HTTP peut causer la modification de données et une opération POST peut causer des modifications à une ressource, autres que celles spécifiées dans la demande. Pour cette raison, il est fourni un spécificateur de référence d'URI Action séparé.

8.7.1.4 Permissions de fichier UNIX

URI: `urn:oasis:names:tc:SAML:1.0:action:unix`

Les actions définies sont l'ensemble des permissions d'accès à des fichiers UNIX exprimées dans la notation numérique (octale).

La chaîne d'action est un code numérique à quatre chiffres:

extended user group world

où la permission d'accès *extended* a la valeur:

+2 si sgid est mis

+4 si suid est mis

Les permissions d'accès *user group* et *world* ont la valeur

+1 si la permission d'exécuter est accordée

+2 si la permission d'écrire est accordée

+4 si la permission de lire est accordée

Par exemple, 0754 note la permission d'accès de fichier UNIX: l'utilisateur lit, écrit, et exécute; le groupe lit et exécute; et le reste du monde lit.

8.7.2 Identifiants de format de nom d'attribut

Les identifiants suivants peuvent être utilisés dans l'attibut NameFormat défini sur le type complexe **AttributeType** pour se référer à la classification des noms d'attribut pour les besoins de l'interprétation du nom.

8.7.2.1 Unspecified

URI: `urn:oasis:names:tc:SAML:2.0:attrname-format:unspecified`

L'interprétation du nom de l'attribut est laissée au soin des implémentations individuelles.

8.7.2.2 Référence d'URI

URI: `urn:oasis:names:tc:SAML:2.0:attrname-format:uri`

Le nom d'attribut suit la convention pour les références d'URI, par exemple, comme utilisé pour les identifiants d'attribut dans XACML. L'interprétation du contenu de l'URI ou du schéma de dénomination est spécifique de l'application. Voir au § 11 les profils d'attribut qui utilisent cet identifiant.

8.7.2.3 Basic

URI: `urn:oasis:names:tc:SAML:2.0:attrname-format:basic`

La classe de chaînes acceptables comme nom de l'attribut doit être tirée de l'ensemble des valeurs appartenant au type de primitive **xs:Name**,comme défini au § 3.3.6 de XML Datatypes du W3C. Voir au § 13 les profils d'attribut qui utilisent cet identifiant.

8.7.3 Identifiants de format d'identifiant de nom

Les identifiants suivants peuvent être utilisés dans l'attribut Format des éléments `<NameID>`, `<NameIDPolicy>`, ou `<Issuer>` (voir au § 8.1.2) pour se référer aux formats communs pour le contenu des éléments et leurs règles de traitement associées, s'il en est.

NOTE – Plusieurs identifiants qui étaient déconseillés dans SAML V1.1 ont été rétirés de SAML V2.0.

8.7.3.1 Unspecified

URI: `urn:oasis:names:tc:SAML:1.1:nameid-format:unspecified`

L'interprétation du contenu de l'élément est laissée au soin des implémentations individuelles.

8.7.3.2 Adresse de messagerie électronique

URI: `urn:oasis:names:tc:SAML:1.1:nameid-format:emailAddress`

Indique que le contenu de l'élément est sous la forme d'une adresse de messagerie électronique, à savoir "addr-spec" comme défini au § 3.4.1 de la RFC 2822 de l'IETF. Une addr-spec a la form local-part@domain. Noter qu'une addr-spec n'a pas de phrase (comme un nom commun) devant elle, pas de commentaire (texte entre parenthèses) après, et n'est pas entourée de crochets angulaires "<" et ">".

8.7.3.3 Nom de sujet X.509

URI: `urn:oasis:names:tc:SAML:1.1:nameid-format:X509SubjectName`

Indique que le contenu de l'élément est sous la forme spécifiée pour le contenu de l'élément `<ds:X509SubjectName>` dans W3C Signature. Les développeurs devraient noter que Signature XML du W3C spécifie les règles de codage pour les noms de sujets X.509 qui diffèrent des règles données dans la RFC 2253 de l'IETF.

8.7.3.4 Nom qualifié de domaine Windows

URI: `urn:oasis:names:tc:SAML:1.1:nameid-format:WindowsDomainQualifiedName`

Indique que le contenu de l'élément est un nom qualifié de domaine Windows. Un nom d'utilisateur qualifié de domaine Windows est une chaîne de la forme "DomainName\UserName". Le nom de domaine et le séparateur "\" peuvent être omis.

8.7.3.5 Nom principal Kerberos

URI: `urn:oasis:names:tc:SAML:2.0:nameid-format:kerberos`

Indique que le contenu de l'élément est sous la forme d'un nom principal Kerberos qui utilise le format `name[/instance]@REALM`. La syntaxe, format et caractères admis pour le nom, instance, et domaine sont décrits dans la RFC 1510 de l'IETF.

8.7.3.6 Identifiant d'entité

URI: `urn:oasis:names:tc:SAML:2.0:nameid-format:entity`

Indique que le contenu de l'élément est l'identifiant d'une entité qui fournit des services fondés sur SAML (tels qu'une autorité SAML, un demandeur, ou répondant) ou est un participant à des profils SAML (tel qu'un fournisseur de service prenant en charge le profil SSO du navigateur). Un tel identifiant peut être utilisé dans l'élément `<Issuer>` pour identifier le producteur d'une demande, réponse, ou assertion SAML, ou au sein de l'élément `<NameID>` pour faire des assertions sur des entités système qui peuvent produire des demandes, réponses, et assertions SAML. Il peut aussi être utilisé dans d'autres éléments et attributs dont l'objet est d'identifier une entité système dans divers échanges de protocole.

La syntaxe d'un tel identifiant est un URI d'une longueur inférieure ou égale à 1024 caractères. Il est recommandé qu'une entité système utilise un URL contenant son propre nom de domaine pour s'identifier.

Les attributs `NameQualifier`, `SPNameQualifier`, et `SPProvidedID` doivent être omis.

8.7.3.7 Identifiant persistant

URI: `urn:oasis:names:tc:SAML:2.0:nameid-format:persistent`

Indique que le contenu de l'élément est un identifiant opaque persistant pour un principal spécifique d'un fournisseur d'identité et d'un fournisseur de service ou affiliation de fournisseurs de service. Les identifiants de nom persistants générés par des fournisseurs d'identité doivent être construits en utilisant des valeurs pseudo-aléatoires qui n'aient pas de correspondance discernable avec l'identifiant réel du sujet (par exemple, un nom d'utilisateur). L'intention est de créer un pseudonyme apparié non-public, pour empêcher la découverte de l'identité ou des activités du sujet. Les valeurs d'identifiants de nom persistants ne doivent pas dépasser 256 caractères.

L'attribut `NameQualifier` de l'élément, s'il est présent, doit contenir l'identifiant unique du fournisseur d'identité qui a généré l'identifiant (voir au § 8.7.3.6). Il peut être omis si la valeur peut être déduite du contexte du message qui contient l'élément, tel que le producteur d'un message de protocole ou d'une assertion contenant l'identifiant dans son sujet. Une entité système différente pourrait ultérieurement produire son propre message de protocole ou assertion contenant l'identifiant; l'attribut `NameQualifier` ne change pas dans ce cas, mais doit continuer à identifier l'entité qui a créé l'identifiant (et ne doit pas être omis dans ce cas).

L'attribut `SPNameQualifier` de l'élément, s'il est présent, doit contenir l'identifiant unique du fournisseur de service ou de l'affiliation de fournisseurs pour lequel l'identifiant a été généré (voir au § 8.7.3.6). Il peut être omis si l'élément est

contenu dans un message destiné seulement à la consommation directe du fournisseur de service, et sa valeur serait l'identifiant unique de ce fournisseur de service.

L'attribut `SPProvidedID` de l'élément doit contenir l'identifiant de remplacement du principal le plus récemment établi par le fournisseur de service ou l'affiliation, s'il en est (voir au § 8.2.6). Si un tel identifiant n'a pas été établi, l'attribut doit alors être omis.

Les identifiants persistants sont destinés à constituer un mécanisme de protection de la confidentialité; à ce titre, ils ne doivent pas être partagés en clair avec des fournisseurs autres que ceux qui ont établi l'identifiant partagé. De plus, ils ne doivent pas apparaître dans les fichiers de journalisation ou localisations similaires sans contrôles et protections appropriés. Les mises en œuvre qui n'ont pas ces exigences ont toute liberté pour utiliser d'autres types d'identifiants dans leurs échanges SAML, mais les producteurs d'assertions surchargent ce format avec des valeurs persistantes mais non-opaques.

Alors que les identifiants persistants sont normalement utilisés pour refléter une relation comptable entre une paire de fournisseurs, un fournisseur de service n'est pas obligé de reconnaître ou faire usage de la nature à long terme de l'identifiant persistant ou d'établir une telle liaison. Une telle relation "unilatérale" n'est pas notablement différente et n'affecte pas le comportement du fournisseur d'identité ou des règles de traitement spécifiques des identifiants persistants dans les protocoles définis dans la présente Recommandation.

Les attributs `NameQualifier` et `SPNameQualifier` indiquent le caractère directionnel de la création, mais non son utilisation. Si un identifiant persistant est créé par un fournisseur d'identité particulier, la valeur d'attribut `NameQualifier` est établie de façon permanente à ce moment. Si un fournisseur de service qui reçoit un tel identifiant prend le rôle de fournisseur d'identité et produit ses propres assertions contenant cet identifiant, la valeur d'attribut `NameQualifier` ne change pas (et ne serait bien sûr pas omise). Il peut aussi choisir à la place de créer son propre identifiant persistant pour représenter le principal et lier les deux valeurs. C'est une décision qui appartient à la mise en œuvre.

8.7.3.8 Identifiant transitoire

URI: `urn:oasis:names:tc:SAML:2.0:nameid-format:transient`

Indique que le contenu de l'élément est un identifiant avec une sémantique transitoire et devrait être traité comme une valeur opaque et temporaire par le consommateur d'assertions. Les valeurs d'identifiant transitoires doivent être générées conformément aux règles des identifiants SAML (voir au § 7.4), et ne doivent pas dépasser une longueur de 256 caractères.

Les attributs `NameQualifier` et `SPNameQualifier` peuvent être utilisés pour signifier que l'identifiant représente un identifiant apparié transitoire et temporaire. Dans un tel cas, ils peuvent être omis, conformément aux règles spécifiées au paragraphe 8.7.3.7.

8.7.4 Identifiants de consentement

Les identifiants suivants peuvent être utilisés dans l'attribut `Consent` défini sur les types complexes **RequestAbstractType** et **StatusResponseType** pour communiquer si un principal a donné son consentement, et sous quelles conditions, pour le message.

8.7.4.1 Unspecified

URI: `urn:oasis:names:tc:SAML:2.0:consent:unspecified`

Aucune revendication n'est faite quant au consentement du principal.

8.7.4.2 Obtained

URI: `urn:oasis:names:tc:SAML:2.0:consent:obtained`

`Indique` que le consentement d'un principal a été obtenu par le producteur du message.

8.7.4.3 Prior

URI: `urn:oasis:names:tc:SAML:2.0:consent:prior`

Indique que le consentement d'un principal a été obtenu par le producteur du message à un certain moment avant l'action qui a initialisé le message.

8.7.4.4 Implicit

URI: `urn:oasis:names:tc:SAML:2.0:consent:current-implicit`

Indique que le consentement d'un principal a été obtenu implicitement par le producteur du message durant l'action qui a initialisé le message, au titre d'une indication de consentement plus large. Le consentement implicite est normalement plus proche de l'action dans le temps et la présentation que le consentement préalable, comme une partie d'une session d'activités.

8.7.4.5 Explicit

URI: `urn:oasis:names:tc:SAML:2.0:consent:current-explicit`

Indique que le consentement d'un principal a été obtenu explicitement par le producteur du message durant l'action qui a initialisé le message.

8.7.4.6 Unavailable

URI: `urn:oasis:names:tc:SAML:2.0:consent:unavailable`

Indique que le producteur du message n'a pas obtenu de consentement.

8.7.4.7 Inapplicable

URI: `urn:oasis:names:tc:SAML:2.0:consent:inapplicable`

Indique que le producteur du message pense qu'il n'est pas besoin d'obtenir ou de rapporter le consentement.

9 Métadonnées SAML

Les profils SAML exigent des accords entre entités système au sujet des identifiants, de la prise en charge des liaisons et des points d'extrémité, des certificats et des clés, et ainsi de suite. Le présent paragraphe définit un format de métadonnées extensible pour les entités système SAML, organisé par rôles qui reflètent les profils SAML. De tels rôles incluent celui de fournisseur d'identité SSO, de fournisseur de service SSO, d'affiliation, d'autorité d'attribut, de demandeur d'attribut, et de point de décision de politique.

9.1 Métadonnées

Les métadonnées SAML sont organisées autour de collections extensibles de rôles représentant des combinaisons communes de protocoles et profils SAML pris en charge par les entités système. Chaque rôle est décrit par un élément dérivé d'un type de base extensible de `RoleDescriptor`. De tels descripteurs sont à leur tour collectés dans l'élément contenant `<EntityDescriptor>`, la principale unité de métadonnées SAML. Une entité peut aussi représenter une affiliation d'autres entités, telle qu'une affiliation de fournisseurs de service. Le `<AffiliationDescriptor>` est fourni à cette fin.

De tels descripteurs peuvent à leur tour être agrégés en groupes enchassés en utilisant l'élément `<EntitiesDescriptor>`.

Divers mécanismes de sécurité pour établir la confiance dans les métadonnées peuvent être pris en charge, en particulier avec la capacité à signer individuellement la plupart des éléments définis dans la présente Recommandation.

Lorsque des éléments avec une relation parent/enfant contiennent des attributs communs, tels que la mémoire cache ou des informations sur l'expiration, l'élément parent a priorité.

> NOTE – En général, les métadonnées SAML ne sont pas à prendre comme une déclaration autoritaire sur les capacités ou options d'une entité système donnée. C'est-à-dire que, alors qu'elles devraient être exactes, il n'est pas nécessaire qu'elles soient exhaustives. L'omission d'une option particulière n'implique pas qu'elle soit ou non prise en charge, et simplement qu'elle n'est pas revendiquée. A titre d'exemple, un attribut d'autorité SAML peut prendre en charge un nombre quelconque d'attributs non nommés dans `<AttributeAuthorityDescriptor>`. Les omissions peuvent refléter la confidentialité ou un certain nombre d'autres considérations. A l'inverse, indiquer la prise en charge d'un attribut donné n'implique pas qu'un demandeur donné va ou veut le recevoir.

9.1.1 Espaces de nom

Les métadonnées SAML utilisent les espaces de nom suivants:

```
urn:oasis:names:tc:SAML:2.0:metadata
```

La présente Recommandation utilise le préfixe d'espace de nom `md`: pour se référer à l'espace de nom ci-dessus.

Le fragment de schéma suivant illustre l'utilisation des espaces de nom dans les documents de métadonnées SAML:

```
<schema
    targetNamespace="urn:oasis:names:tc:SAML:2.0:metadata"
    xmlns:md="urn:oasis:names:tc:SAML:2.0:metadata"
    xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
    xmlns:xenc="http://www.w3.org/2001/04/xmlenc#"
    xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion"
    xmlns="http://www.w3.org/2001/XMLSchema"
    elementFormDefault="unqualified"
    attributeFormDefault="unqualified"
    blockDefault="substitution"
    version="2.0">
    <import namespace="http://www.w3.org/2000/09/xmldsig#"
        schemaLocation="http://www.w3.org/TR/2002/REC-xmldsig-core-
20020212/xmldsig-core-schema.xsd"/>
    <import namespace="http://www.w3.org/2001/04/xmlenc#"
        schemaLocation="http://www.w3.org/TR/2002/REC-xmlenc-core-
20021210/xenc-schema.xsd"/>
    <import namespace="urn:oasis:names:tc:SAML:2.0:assertion"
        schemaLocation="saml-schema-assertion-2.0.xsd"/>
    <import namespace="http://www.w3.org/XML/1998/namespace"
        schemaLocation="http://www.w3.org/2001/xml.xsd"/>
    <annotation>
        <documentation>
            Document identifier: saml-schema-metadata-2.0
            Location: http://docs.oasis-open.org/security/saml/v2.0/
            Revision history:
              V2.0 (March, 2005):
                Schema for SAML metadata, first published in SAML 2.0.
        </documentation>
    </annotation>
…
</schema>
```

9.1.2 Types communs

Le présent paragraphe définit plusieurs types de métadonnées à utiliser pour définir les éléments et attributs.

9.1.2.1 Type simple entityIDType

Le type simple **entityIDType** restreint le type de données de schéma XML **anyURI** à une longueur maximum de 1024 caractères. **entityIDType** est utilisé comme identifiant unique pour les entités SAML. Voir aussi au § 8.7.3.6. Un identifiant de ce type doit être unique à travers toutes les entités qui interagissent au sein d'un développement donné. Utiliser un URI et s'en tenir à la règle qu'un seul URI ne doit pas se référer à différentes entités satisfait cette exigence.

Le fragment de schéma suivant définit le type simple **entityIDType**:

```
<simpleType name="entityIDType">
      <restriction base="anyURI">
            <maxLength value="1024"/>
      </restriction>
</simpleType>
```

9.1.2.2 Type complexe EndpointType

Le type complexe **EndpointType** décrit un point d'extrémité de liaison de protocole SAML auquel une entité SAML peut envoyer les messages de protocole. Divers éléments de protocole ou de métadonnées spécifiques du profil sont liés à ce type. Il comporte les attributs suivants:

– `Binding` [Exigé]

  Attribut exigé qui spécifie la liaison SAML prise en charge par le point d'extrémité. Chaque liaison a un URI alloué pour l'identifier.

– `Location` [Exigé]

  Attribut d'URI exigé qui spécifie la localisation du point d'extrémité. La syntaxe admissible pour cet URI dépend de la liaison de protocole.

– `ResponseLocation` [Facultatif]

Spécifie facultativement une localisation différente à laquelle les messages de réponse envoyés au titre du protocole ou profil devraient être envoyés. La syntaxe admissible pour cet URI dépend de la liaison de protocole.

L'attribut `ResponseLocation` sert à activer différents points d'extrémité comme récepteurs de messages de demande et de réponse associés à un protocole ou profil, mais pas comme moyen d'équilibrage de charge ou de redondance (plusieurs éléments de ce type peuvent être inclus à cette fin). Lorsqu'un rôle contient un élément de ce type relevant d'un protocole ou profil pour lequel un seul type de message (demande ou réponse) est applicable, l'attribut `ResponseLocation` n'est pas utilisé.

NOTE (informative) – PE41 (voir OASIS PE:2006) précise l'alinéa ci-dessus en ajoutant au texte la phrase suivante:

> Si l'attribut `ResponseLocation` est omis, tout message de réponse associé à un protocole ou profil peut être supposé traité à l'URI indiqué par l'attribut `Location`.

Dans la plupart des contextes, un élément de ce type apparaît dans des séquences non limitées dans le schéma. Ceci est destiné à permettre à un protocole ou profil d'être offert par une entité à plusieurs points d'extrémité, normalement avec différentes liaisons de protocole, permettant au consommateur de métadonnées de choisir un point d'extrémité approprié pour ses besoins. Plusieurs points d'extrémité peuvent aussi offrir l'équilibrage de charge ou la reprise sur défaillance "côté client", en particulier dans le cas d'une liaison de protocole synchrone.

Cet élément permet aussi l'utilisation d'éléments et attributs arbitraires définis dans un espace de nom non-SAML. Un tel contenu doit être qualifié comme espace de nom.

Le fragment de schéma suivant définit le type complexe **EndpointType**:

```
<complexType name="EndpointType">
       <sequence>
              <any namespace="##other" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
       </sequence>
       <attribute name="Binding" type="anyURI" use="required"/>
       <attribute name="Location" type="anyURI" use="required"/>
       <attribute name="ResponseLocation" type="anyURI" use="optional"/>
       <anyAttribute namespace="##other" processContents="lax"/>
</complexType>
```

9.1.2.3 Type complexe IndexedEndpointType

Le type complexe **IndexedEndpointType** étend **EndpointType** avec une paire d'attributs pour permettre l'indexation de points d'extrémité par ailleurs identiques afin qu'ils puissent être référencés par les messages de protocole. Il consiste en les attributs supplémentaires suivants:

– `index` [Exigé]

Attribut exigé qui alloue une valeur d'entier unique au point d'extrémité de façon à ce qu'il puisse être référencé dans un message de protocole. La valeur d'indice a seulement besoin d'être unique au sein d'une collection d'éléments semblables contenus dans le même élément parent (c'est-à-dire, ils n'ont pas besoin d'être uniques sur l'ensemble de l'instance).

– `isDefault` [Facultatif]

Attribut booléen utilisé pour désigner le point d'extrémité par défaut parmi un ensemble indexé. S'il est omis, la valeur est supposée être fausse.

Dans une telle séquence de points d'extrémité fondés sur ce type, le point d'extrémité est le premier de tels points pour lequel l'attribut `isDefault` est mis à `vrai`. Si de tels points n'existent pas, le point d'extrémité par défaut est le premier de tels points sans l'attribut `isDefault` mis à `faux`. Si de tels points d'extrémité n'existent pas, le point d'extrémité par défaut est le premier élément dans la séquence.

NOTE (informative) – PE37 (voir OASIS PE:2006) suggère de préciser l'alinéa ci-dessus par:

> Dans toute séquence de points d'extrémité indexés de la sorte, qui partagent un nom d'élément et un espace de nom communs (c'est-à-dire toutes les instances de `<md:AssertionConsumerService>` dans un rôle), le point d'extrémité par défaut est le premier de tels points dont l'attribut `isDefault` est mis à `vrai`. S'il n'existe pas de tel point, le point par défaut est le premier dont l'attribut `isDefault` est mis à `faux`. Si un tel points n'existe pas, le point d'extrémité par défaut est le premier élément dans la séquence.

Le fragment de schéma suivant définit le type complexe **IndexedEndpointType**:

```
<complexType name="IndexedEndpointType">
      <complexContent>
            <extension base="md:EndpointType">
                  <attribute name="index" type="unsignedShort"
use="required"/>
                  <attribute name="isDefault" type="boolean"
use="optional"/>
            </extension>
      </complexContent>
</complexType>
```

9.1.2.4 Type complexe localizedNameType

Le type complexe **localizedNameType** étend un élément valorisé par une chaîne avec un attribut de langage XML standard. Le fragment de schéma suivant définit le type complexe **localizedNameType**:

```
<complexType name="localizedNameType">
      <simpleContent>
            <extension base="string">
                  <attribute ref="xml:lang" use="required"/>
            </extension>
      </simpleContent>
</complexType>
```

9.1.2.5 Type complexe localizedURIType

Le type complexe **localizedURIType** étend un élément valorisé par un URI avec un attribut de langage XML standard.

Le fragment de schéma suivant définit le type complexe **localizedURIType**:

```
<complexType name="localizedURIType">
    <simpleContent>
            <extension base="anyURI">
                  <attribute ref="xml:lang" use="required"/>
            </extension>
      </simpleContent>
</complexType>
```

9.1.3 Éléments racine

Une instance de métadonnées SAML décrit une entité seule ou plusieurs entités. Dans le premier cas, l'élément racine doit être `<EntityDescriptor>`. Dans le second cas, l'élément racine doit être `<EntitiesDescriptor>`.

9.1.3.1 Élément <EntitiesDescriptor>

L'élément `<EntitiesDescriptor>` contient les métadonnées pour un groupe facultativement désigné d'entités SAML. Son type complexe **EntitiesDescriptorType** contient une séquence d'éléments `<EntityDescriptor>`, d'éléments `<EntitiesDescriptor>`, ou les deux:

– `ID` [Facultatif]

  Identifiant unique par document pour l'élément, normalement utilisé comme point de référence à la signature.

– `validUntil` [Facultatif]

  Attribut facultatif qui indique l'heure d'expiration des métadonnées contenues dans l'élément et tout élément contenu.

– `cacheDuration` [Facultatif]

  Attribut facultatif qui indique la durée maximale pendant laquelle un consommateur devrait mettre en mémoire cache les métadonnées contenues dans l'élément et tout élément contenu.

– `Name` [Facultatif]

  Chaîne de nom qui identifie un groupe d'entités SAML dans le contexte d'un développement.

– `<ds:Signature>` [Facultatif]

  Signature XML qui authentifie l'élément contenant et son contenu, comme décrit au § 8.

– `<Extensions>` [Facultatif]

  Contient les extensions de métadonnées facultatives qui sont convenues entre un éditeur et un consommateur de métadonnées. Les éléments d'extension doivent être qualifiés en espace de nom par un espace de nom non défini par SAML.

– `<EntitiesDescriptor>` ou `<EntityDescriptor>` [Un ou plusieurs]

  Contient les métadonnées pour une ou plusieurs entités SAML, ou un groupe enchassé de métadonnées supplémentaires.

Lorsqu'utilisé comme élément racine d'une instance de métadonnées, cet élément doit contenir un attribut `validUntil` ou `cacheDuration`. Il est recommandé que seul l'élément racine d'une instance de métadonnées contienne l'un et l'autre attribut.

Le fragment de schéma suivant définit l'élément `<EntitiesDescriptor>` et son type complexe **EntitiesDescriptorType**:

```
<element name="EntitiesDescriptor" type="md:EntitiesDescriptorType"/>
<complexType name="EntitiesDescriptorType">
       <sequence>
              <element ref="ds:Signature" minOccurs="0"/>
              <element ref="md:Extensions" minOccurs="0"/>
              <choice minOccurs="1" maxOccurs="unbounded">
                     <element ref="md:EntityDescriptor"/>
                     <element ref="md:EntitiesDescriptor"/>
              </choice>
       </sequence>
       <attribute name="validUntil" type="dateTime" use="optional"/>
       <attribute name="cacheDuration" type="duration" use="optional"/>
       <attribute name="ID" type="ID" use="optional"/>
       <attribute name="Name" type="string" use="optional"/>
</complexType>
<element name="Extensions" type="md:ExtensionsType"/>
<complexType final="#all" name="ExtensionsType">
       <sequence>
              <any namespace="##other" processContents="lax"
maxOccurs="unbounded"/>
       </sequence>
</complexType>
```

9.1.3.2 Élément <EntityDescriptor>

L'élément `<EntityDescriptor>` spécifie des métadonnées pour une seule entité SAML. Une seule entité peut jouer de nombreux rôles différents dans la prise en charge de nombreux profils. La présente Recommandation prend en charge directement les rôles concrets suivants ainsi que l'élément abstrait `<RoleDescriptor>` pour l'extensibilité:

- fournisseur d'identité SSO;
- fournisseur de service SSO;
- autorité d'authentification;
- autorité d'attribut;
- point de décision de politique;
- affiliation.

Son type complexe **EntityDescriptorType** comporte les éléments et attributs suivants:

– `entityID` [Exigé]

  Spécifie l'identifiant unique de l'entité SAML dont les métadonnées sont décrites par le contenu de l'élément.

– `ID` [Facultatif]

  Identifiant unique par document pour l'élément, normalement utilisé comme point de référence à la signature.

– `validUntil` [Facultatif]

  Attribut facultatif qui indique l'instant d'expiration des métadonnées contenues dans l'élément et tout élément contenu.

– `cacheDuration` [Facultatif]

  Attribut facultatif qui indique la durée maximale pendant laquelle un consommateur devrait conserver en mémoire cache les métadonnées contenues dans l'élément et tout élément contenu.

– `<ds:Signature>` [Facultatif]

  Signature XML qui authentifie l'élément contenant et son contenu.

– `<Extensions>` [Facultatif]

  Contiennent les extensions de métadonnées facultatives qui sont convenues entre un éditeur et un consommateur de métadonnées. Les éléments d'extension doivent être qualifiés en espace de nom par une espace de nom non défini par SAML.

– `<RoleDescriptor>`, `<IDPSSODescriptor>`, `<SPSSODescriptor>`, `<AuthnAuthorityDescriptor>`, `<AttributeAuthorityDescriptor>`, `<PDPDescriptor>` [Un ou plusieurs] ou

– `<AffiliationDescriptor>` [Exigé]

  Le contenu principal de l'élément est soit une séquence d'un ou plusieurs éléments descripteurs de rôle, soit un descripteur spécialisé qui définit une affiliation.

– `<Organization>` [Facultatif]

  Élément facultatif qui identifie l'organisation responsable pour l'entité SAML décrite par l'élément.

– `<ContactPerson>` [Zéro, un ou plusieurs]

  Séquence facultative d'éléments qui identifient diverses sortes de contacts personnels.

– `<AdditionalMetadataLocation>` [Zéro, un ou plusieurs]

  Séquence facultative de localisations qualifiées en espace de nom où des métadonnées supplémentaires existent pour l'entité SAML. Cela peut inclure des métadonnées dans des formats de remplacement ou décrivant l'adhésion à d'autres Recommandations non-SAML.

Des attributs arbitraires qualifiés en espace de nom provenant d'espaces de nom non définis par SAML peuvent aussi être inclus.

Lorsqu'utilisé comme élément racine d'une instance de métadonnées, cet élément doit contenir un attribut `validUntil` ou un attribut `cacheDuration`. Il est recommandé que seul l'élément racine de l'instance de métadonnées contienne l'un et l'autre attributs.

Il est recommandé que si plusieurs éléments de description de rôle du même type apparaissent, ils n'aient pas de valeurs de `protocolSupportEnumeration` qui se chevauchent. Le choix à partir de plusieurs éléments de description de rôle du même type qui partagent une valeur de `protocolSupportEnumeration` est indéfini dans la présente Recommandation, mais peut être défini par des profils de métadonnées, éventuellement grâce à l'utilisation d'attributs d'extension distinctifs.

Le fragment de schéma suivant définit l'élément `<EntityDescriptor>` et son type complexe **EntityDescriptorType**:

```
<element name="EntityDescriptor" type="md:EntityDescriptorType"/>
<complexType name="EntityDescriptorType">
       <sequence>
              <element ref="ds:Signature" minOccurs="0"/>
              <element ref="md:Extensions" minOccurs="0"/>
              <choice>
                     <choice maxOccurs="unbounded">
                            <element ref="md:RoleDescriptor"/>
                            <element ref="md:IDPSSODescriptor"/>
                            <element ref="md:SPSSODescriptor"/>
                            <element ref="md:AuthnAuthorityDescriptor"/>
                            <element ref="md:AttributeAuthorityDescriptor"/>
                            <element ref="md:PDPDescriptor"/>
                     </choice>
                     <element ref="md:AffiliationDescriptor"/>
              </choice>
              <element ref="md:Organization" minOccurs="0"/>
              <element ref="md:ContactPerson" minOccurs="0"
maxOccurs="unbounded"/>
```

```
              <element ref="md:AdditionalMetadataLocation" minOccurs="0"
maxOccurs="unbounded"/>
       </sequence>
       <attribute name="entityID" type="md:entityIDType" use="required"/>
       <attribute name="validUntil" type="dateTime" use="optional"/>
       <attribute name="cacheDuration" type="duration" use="optional"/>
       <attribute name="ID" type="ID" use="optional"/>
       <anyAttribute namespace="##other" processContents="lax"/>
</complexType>
```

9.1.3.2.1 Élément <Organization>

L'élément `<Organization>` spécifie des informations de base sur une organisation responsable d'une entité ou rôle SAML. L'utilisation de cet élément est toujours facultative. Son contenu est informatif par nature et ne correspond pas directement à un élément ou attribut SAML central. Son type complexe **OrganizationType** comporte les éléments suivants:

– `<Extensions>` [Facultatif]

  Contient des extensions de métadonnées facultatives qui sont convenues entre un éditeur et un consommateur de métadonnées. Les extensions ne doivent pas inclure d'éléments globaux (non qualifiés d'espace de nom) ou d'éléments qualifiés par un espace de nom défini par SAML au sein de cet élément.

– `<OrganizationName>` [Un ou plusieurs]

  Un ou plusieurs noms qualifiés en langage qui peuvent être ou non lisibles par l'homme.

– `<OrganizationDisplayName>` [Un ou plusieurs]

  Un ou plusieurs noms qualifiés en langage qui peuvent être ou non lisibles par l'homme.

– `<OrganizationURL>` [Un ou plusieurs]

  Un ou plusieurs URI qualifiés en langage qui spécifient une localisation sur laquelle diriger un utilisateur pour des informations supplémentaires. Le qualificatif de langage se réfère au contenu du matériel à la localisation spécifiée.

Des attributs d'espace de nom qualifiés arbitraires proveant d'espaces de nom non définis par SAML peuvent aussi être inclus.

Le fragment de schéma suivant définit l'élément `<Organization>` et son type complexe **OrganizationType**:

```
<element name="Organization" type="md:OrganizationType"/>
<complexType name="OrganizationType">
       <sequence>
              <element ref="md:Extensions" minOccurs="0"/>
              <element ref="md:OrganizationName" maxOccurs="unbounded"/>
              <element ref="md:OrganizationDisplayName"
maxOccurs="unbounded"/>
              <element ref="md:OrganizationURL" maxOccurs="unbounded"/>
       </sequence>
       <anyAttribute namespace="##other" processContents="lax"/>
</complexType>
<element name="OrganizationName" type="md:localizedNameType"/>
<element name="OrganizationDisplayName" type="md:localizedNameType"/>
<element name="OrganizationURL" type="md:localizedURIType"/>
```

9.1.3.2.2 Élément <ContactPerson>

L'élément `<ContactPerson>` spécifie des informations de contact de base sur la personne responsable dans une certaine mesure d'une entité ou rôle SAML. L'utilisation de cet élément est toujours facultative. Son contenu est informatif par nature et ne correspond directement à aucun élément ou attribut SAML central. Son type complexe **ContactType** comporte les éléments et attributs suivants:

– `contactType` [Exigé]

  Spécifie le type de contact qui utilise l'énumération **ContactTypeType**. Les valeurs possibles sont technical, support, administrative, billing, et other.

– `<Extensions>` [Facultatif]

  Contient les extensions de métadonnées facultatives qui sont convenues entre un éditeur et un consommateur de métadonnées. Les éléments d'extension doivent être qualifiés en espace de nom par un espace de nom non défini par SAML.

– `<Company>` [Facultatif]

  Élément de chaîne facultatif qui spécifie le nom de la compagnie de la personne à contacter.

– `<GivenName>` [Facultatif]

  Élément de chaîne facultatif qui spécifie le prénom de la personne à contacter.

– `<SurName>` [Facultatif]

  Élément de chaîne facultatif qui spécifie le nom de famille de la personne à contacter.

– `<EmailAddress>` [Zéro, un ou plusieurs]

  Zéro un ou plusieurs éléments contenant des URI mailto: qui représentent les adresses de messagerie électronique qui appartiennent à la personne à contacter.

– `<TelephoneNumber>` [Zéro, un ou plusieurs]

  Zéro un ou plusieurs éléments de chaîne qui spécifient un numéro de téléphone de la personne à contacter.

Des attributs d'espace de nom qualifié arbitraire provenant d'espaces de nom non définis par SAML peuvent aussi être inclus.

Le fragment de schéma suivant définit l'élément `<ContactPerson>` et son type complexe **ContactType**:

```
<element name="ContactPerson" type="md:ContactType"/>
<complexType name="ContactType">
       <sequence>
              <element ref="md:Extensions" minOccurs="0"/>
              <element ref="md:Company" minOccurs="0"/>
              <element ref="md:GivenName" minOccurs="0"/>
              <element ref="md:SurName" minOccurs="0"/>
              <element ref="md:EmailAddress" minOccurs="0"
maxOccurs="unbounded"/>
              <element ref="md:TelephoneNumber" minOccurs="0"
maxOccurs="unbounded"/>
       </sequence>
       <attribute name="contactType" type="md:ContactTypeType"
use="required"/>
       <anyAttribute namespace="##other" processContents="lax"/>
</complexType>
<element name="Company" type="string"/>
<element name="GivenName" type="string"/>
<element name="SurName" type="string"/>
<element name="EmailAddress" type="anyURI"/>
<element name="TelephoneNumber" type="string"/>
<simpleType name="ContactTypeType">
       <restriction base="string">
              <enumeration value="technical"/>
              <enumeration value="support"/>
              <enumeration value="administrative"/>
              <enumeration value="billing"/>
              <enumeration value="other"/>
       </restriction>
</simpleType>
```

9.1.3.2.3 Élément <AdditionalMetadataLocation>

L'élément `<AdditionalMetadataLocation>` est un URI d'espace de nom qualifié qui spécifie où peuvent exister des métadonnées supplémentaires fondées sur XML pour une entité SAML. Son type complexe **AdditionalMetadataLocationType** étend le type **anyURI** avec un attribut d'espace de nom (aussi de type **anyURI**). Cet attribut exigé doit contenir l'espace de nom XML de l'élément racine de l'instance de document trouvé à la localisation spécifiée.

Le fragment de schéma suivant définit l'élément <AdditionalMetadataLocation> et son type complexe **AdditionalMetadataLocationType**:

```
<element name="AdditionalMetadataLocation"
type="md:AdditionalMetadataLocationType"/>
<complexType name="AdditionalMetadataLocationType">
       <simpleContent>
              <extension base="anyURI">
                     <attribute name="namespace" type="anyURI"
use="required"/>
              </extension>
       </simpleContent>
</complexType>
```

9.1.4 Éléments de descripteur de rôle

Les éléments du présent paragraphe constituent les composants bruts de la prise en charge opérationnelle des métadonnées. Chaque élément (sauf d'élément abstrait) définit une collection spécifique de comportements opérationnels dans la prise en charge des profils SAML.

9.1.4.1 Élément <RoleDescriptor>

L'élément <RoleDescriptor> est un point d'extension abstrait qui contient des informations descriptives communes destinées à fournir une normalisation du traitement des différents rôles. De nouveaux rôles peuvent être définis en étendant son type complexe abstrait **RoleDescriptorType**, qui contient les éléments et attributs suivants:

– ID [Facultatif]

  Identifiant unique par document, normalement utilisé comme point de référence lors de la signature.

– validUntil [Facultatif]

  Attribut facultatif qui indique l'heure d'expiration des métadonnées contenues dans l'élément et tout élément contenu.

– cacheDuration [Facultatif]

  Attribut facultatif qui indique la durée maximale pendant laquelle un consommateur devrait conserver en mémoire cache les métadonnées contenues dans l'élément et tout élément contenu.

– protocolSupportEnumeration [Exigé]

  Ensemble d'URI délimités par des espaces blancs qui identifient l'ensemble de spécifications de protocole pris en charge par l'élément de rôle. Pour les entitiés SAML V2.0, cet ensemble doit inclure l'URI d'espace de nom de protocole SAML, urn:oasis:names:tc:SAML:2.0:protocol. Les Recommandations ultérieures de SAML partagent le même URI d'espace de nom, mais devraient fournir d'autres identifiants de "prise en charge du protocole" pour assurer la discrimination lorsque nécessaire.

– errorURL [Facultatif]

  Attribut d'URI facultatif qui spécifie une localisation pour diriger un utilisateur sur la résolution de problèmes et autre soutien en rapport avec ce rôle.

– <ds:Signature> [Facultatif]

  Signature XML qui authentifie les éléments contenants et leur contenu.

– <Extensions> [Facultatif]

  Contient les extensions de métadonnées facultatives qui sont convenues entre un éditeur et un consommateur de métadonnées. Les éléments d'extension doivent être des espaces de nom qualifiés par un espace de nom non défini par SAML.

– <KeyDescriptor> [Zéro, un ou plusieurs]

  Séquence facultative d'éléments qui fournit des informations sur les clés cryptographiques qu'utilise l'entité lorsqu'elle joue ce rôle.

– <Organization> [Facultatif]

  Élément facultatif qui spécifie l'organisation associée à ce rôle. Identique à l'élément utilisé au sein de l'élément <EntityDescriptor>.

– `<ContactPerson>` [Zéro, un ou plusieurs]

  Séquence facultative d'éléments qui spécifie les contacts associés à ce rôle. Identique à l'élément utilisé au sein de l'élément `<EntityDescriptor>`.

Des attributs d'espace de nom qualifié arbitraire provenant d'espaces de nom non définis par SAML peuvent aussi être inclus.

Le fragment de schéma suivant définit l'élément `<RoleDescriptor>` et son type complexe **RoleDescriptorType**:

```
<element name="RoleDescriptor" type="md:RoleDescriptorType"/>
<complexType name="RoleDescriptorType" abstract="true">
       <sequence>
              <element ref="ds:Signature" minOccurs="0"/>
              <element ref="md:Extensions" minOccurs="0"/>
              <element ref="md:KeyDescriptor" minOccurs="0"
maxOccurs="unbounded"/>
              <element ref="md:Organization" minOccurs="0"/>
              <element ref="md:ContactPerson" minOccurs="0"
maxOccurs="unbounded"/>
       </sequence>
       <attribute name="ID" type="ID" use="optional"/>
       <attribute name="validUntil" type="dateTime" use="optional"/>
       <attribute name="cacheDuration" type="duration" use="optional"/>
       <attribute name="protocolSupportEnumeration" type="md:anyURIListType"
use="required"/>
       <attribute name="errorURL" type="anyURI" use="optional"/>
       <anyAttribute namespace="##other" processContents="lax"/>
</complexType>
<simpleType name="anyURIListType">
       <list itemType="anyURI"/>
</simpleType>
```

9.1.4.1.1 Élément <KeyDescriptor>

L'élément `<KeyDescriptor>` fournit des informations sur la ou les clés cryptographiques qu'utilise une entité pour signer des données ou recevoir des clés chiffrées, ainsi que des détails cryptographiques supplémentaires. Son type complexe **KeyDescriptorType** comporte les éléments et attributs suivants:

– `use` [Facultatif]

  Attribut facultatif qui spécifie l'objet de la clé décrite. Les valeurs sont tirées de l'énumération **KeyTypes**, et consistent en valeurs `encryption` (chiffrées) et en `signing` (signatures).

– `<ds:KeyInfo>` [Exigé]

  Élément facultatif qui identifie directement ou indirectement une clé. Voir le document Signatures XML du W3C pour des précisions sur l'utilisation de cet élément.

– `<EncryptionMethod>` [Zéro, un ou plusieurs]

  Élément facultatif qui spécifie un algorithme et des réglages spécifiques de l'algorithme pris en charge par l'entité. Le contenu exact varie sur la base de l'algorithme pris en charge. Voir au document Chiffrement du W3C la définition du type complexe **xenc:EncryptionMethodType** de cet élément.

Le fragment de schéma suivant définit l'élément `<KeyDescriptor>` et son type complexe **KeyDescriptorType**:

```
<element name="KeyDescriptor" type="md:KeyDescriptorType"/>
<complexType name="KeyDescriptorType">
    <sequence>
        <element ref="ds:KeyInfo"/>
        <element ref="md:EncryptionMethod" minOccurs="0"
maxOccurs="unbounded"/>
    </sequence>
    <attribute name="use" type="md:KeyTypes" use="optional"/>
</complexType>
<simpleType name="KeyTypes">
    <restriction base="string">
        <enumeration value="encryption"/>
        <enumeration value="signing"/>
    </restriction>
```

```
</simpleType>
<element name="EncryptionMethod" type="xenc:EncryptionMethodType"/>
```

9.1.4.2 Type complexe SSODescriptorType

Le type abstrait **SSODescriptorType** est un type de base commun pour les types concrets **SPSSODescriptorType** et **IDPSSODescriptorType**, décrits dans les paragraphes suivants. Il étend **RoleDescriptorType** avec des éléments reflétant les profils communs aux fournisseurs d'identité et aux fournisseurs de service qui prennent en charge SSO, et contient les éléments supplémentaires suivants:

– `<ArtifactResolutionService>` [Zéro, un ou plusieurs]

Zéro, un ou plusieurs éléments du type **IndexedEndpointType** qui décrivent des points d'extrémité indexés qui prennent en charge le profil de résolution d'artifice, défini au § 12. L'attribut ResponseLocation doit être omis.

– `<SingleLogoutService>` [Zéro, un ou plusieurs]

Zéro, un ou plusieurs éléments du type **EndpointType** qui décrivent les points d'extrémité qui prennent en charge les profils de terminaison de session unique, définis au § 12.

– `<ManageNameIDService>` [Zéro, un ou plusieurs]

Zéro, un ou plusieurs éléments du type **EndpointType** qui décrivent les points d'extrémité qui prennent en charge les profils de gestion d'identifiant de nom, définis au § 12.

– `<NameIDFormat>` [Zéro, un ou plusieurs]

Zéro, un ou plusieurs éléments du type **anyURI** qui énumèrent les formats d'identifiant de nom pris en charge à cette entité système jouant ce rôle.

Le fragment de schéma suivant définit le type complexe **SSODescriptorType**:

```
<complexType name="SSODescriptorType" abstract="true">
      <complexContent>
            <extension base="md:RoleDescriptorType">
                  <sequence>
                        <element ref="md:ArtifactResolutionService"
minOccurs="0" maxOccurs="unbounded"/>
                        <element ref="md:SingleLogoutService"
minOccurs="0" maxOccurs="unbounded"/>
                        <element ref="md:ManageNameIDService"
minOccurs="0" maxOccurs="unbounded"/>
                        <element ref="md:NameIDFormat" minOccurs="0"
maxOccurs="unbounded"/>
                  </sequence>
            </extension>
      </complexContent>
</complexType>
<element name="ArtifactResolutionService" type="md:IndexedEndpointType"/>
<element name="SingleLogoutService" type="md:EndpointType"/>
<element name="ManageNameIDService" type="md:EndpointType"/>
<element name="NameIDFormat" type="anyURI"/>
```

9.1.4.3 Élément <IDPSSODescriptor>

L'élément `<IDPSSODescriptor>` étend **SSODescriptorType** avec un contenu reflétant les profils spécifiques des fournisseurs d'identité qui prennent en charge SSO. Son type complexe **IDPSSODescriptorType** contient les éléments et attributs supplémentaires suivants:

– `WantAuthnRequestsSigned` [Facultatif]

Attribut facultatif qui indique l'exigence que les messages `<samlp:AuthnRequest>` reçus par ce fournisseur d'identité soient signés. S'il est omis, la valeur est supposée être `faux`.

– `<SingleSignOnService>` [Un ou plusieurs]

Un ou plusieurs éléments du type **EndpointType** qui décrivent les points d'extrémité qui prennent en charge les profils du protocole de demande d'authentification, définis au § 12. Tout fournisseur d'identité prend en charge au moins un tel point d'extrémité, par définition. L'attribut `ResponseLocation` doit être omis.

– `<NameIDMappingService>` [Zéro, un ou plusieurs]

Zéro, un ou plusieurs éléments du type **EndpointType** qui décrivent les points d'extrémité qui prennent en charge le profil de mappage d'identifiant de nom, défini au § 12. L'attribut `ResponseLocation` doit être omis.

– `<AssertionIDRequestService>` [Zéro, un ou plusieurs]

Zéro, un ou plusieurs éléments du type **EndpointType** qui décrivent les points d'extrémité qui prennent en charge le profil du protocole de demande d'assertion ou la liaison d'URI spéciale pour la demande d'assertion définie au § 10.

NOTE 1 (informative) – PE33 (voir OASIS PE:2006) suggère de remplacer protocole de demande d'assertion par interrogation/demande d'assertion.

– `<AttributeProfile>` [Zéro, un ou plusieurs]

Zéro, un ou plusieurs éléments du type **anyURI** qui énumèrent les profils d'attribut pris en charge par ce fournisseur d'identité.

– `<saml:Attribute>` [Zéro, un ou plusieurs]

Zéro, un ou plusieurs éléments qui identifient les attributs SAML pris en charge par le fournisseur d'identité. Des valeurs spécifiques peuvent facultativement être incluses, indiquant que seules certaines valeurs permises par la définition d'attribut sont prises en charge. Dans ce contexte, "pris en charge" pour un attribut signifie que le fournisseur d'identité a la capacité de l'inclure lors de la livraison d'assertions durant l'ouverture unique de session.

NOTE 2 (informative) – PE7 (voir OASIS PE:2006) suggère d'ajouter le texte suivant à la fin de l'alinéa ci-dessus:

L'attribut `WantAuthnRequestsSigned` est destiné à indiquer aux fournisseurs de service si ils peuvent ou non s'attendre à ce qu'un message `<AuthnRequest>` non signé soit accepté par le fournisseur d'identité. Le fournisseur d'identité n'est pas obligé de rejeter une demande non signée et un fournisseur de service n'est pas obligé de signer sa demande, bien qu'il puisse raisonnablement s'attendre à ce qu'une demande non signée soit rejetée. Dans certains cas, un fournisseur de service ne peut même pas savoir quel fournisseur d'identité va finalement recevoir sa demande et y répondre, aussi l'utilisation de cet attribut dans un tel cas ne peut pas être définie de façon stricte. De plus, noter que la méthode spécifique de signature à laquelle on peut s'attendre dépend de la liaison. La liaison Redirect HTTP du § 10.2.4 exige que la signature soit appliquée à la valeur codée en URL plutôt que placée au sein du message XML, alors que les autres liaisons permettent généralement que la signature soit au sein du message de la façon habituelle.

Le fragment de schéma suivant définit l'élément `<IDPSSODescriptor>` et son type complexe **IDPSSODescriptorType**:

```
<element name="IDPSSODescriptor" type="md:IDPSSODescriptorType"/>
<complexType name="IDPSSODescriptorType">
       <complexContent>
              <extension base="md:SSODescriptorType">
                     <sequence>
                            <element ref="md:SingleSignOnService"
maxOccurs="unbounded"/>
                            <element ref="md:NameIDMappingService"
minOccurs="0" maxOccurs="unbounded"/>
                            <element ref="md:AssertionIDRequestService"
minOccurs="0" maxOccurs="unbounded"/>
                            <element ref="md:AttributeProfile" minOccurs="0"
maxOccurs="unbounded"/>
                            <element ref="saml:Attribute" minOccurs="0"
maxOccurs="unbounded"/>
                     </sequence>
                     <attribute name="WantAuthnRequestsSigned"
type="boolean" use="optional"/>
              </extension>
       </complexContent>
</complexType>
<element name="SingleSignOnService" type="md:EndpointType"/>
<element name="NameIDMappingService" type="md:EndpointType"/>
<element name="AssertionIDRequestService" type="md:EndpointType"/>
<element name="AttributeProfile" type="anyURI"/>
```

9.1.4.4 Élément <SPSSODescriptor>

L'élément `<SPSSODescriptor>` étend **SSODescriptorType** avec un contenu qui reflète les profils spécifiques des fournisseurs de service. Son type complexe **SPSSODescriptorType** contient les éléments et attributs supplémentaires suivants:

– `AuthnRequestsSigned` [Facultatif]

Attribut facultatif qui indique si les messages `<samlp:AuthnRequest>` envoyés par ce fournisseur de service seront signés. S'il est omis, la valeur est supposée être `faux`.

NOTE 1 (informative) – PE7 (voir OASIS PE:2006) suggère d'ajouter le texte suivant à la fin de l'alinéa ci-dessus:

Une valeur de faux (ou l'omission de cet attribut) n'implique pas que le fournisseur de service ne signera jamais sa demande ou qu'une demande signée devrait être considérée comme une erreur. Cependant, un fournisseur d'identité qui reçoit un message `<samlp:AuthnRequest>` non signé d'un fournisseur de service dont les métadonnées contiennent cet attribut avec une valeur de `vrai` doit retourner une réponse d'erreur SAML et ne doit pas satisfaire la demande.

– `WantAssertionsSigned` [Facultatif]

Attribut facultatif qui indique l'exigence que les éléments `<saml:Assertion>` reçus par ce fournisseur de service soient signés. S'il est omis, la valeur est supposée être `faux`. Cette exigence s'ajoute à toute exigence de signature déduite de l'utilisation d'une combinaison particulière de profil/liaison.

NOTE 2 (informative) – PE7 (voir OASIS PE:2006) suggère d'ajouter le texte suivant à la fin de l'alinéa ci-dessus:

Noter qu'une signature d'enveloppe à la liaison SAML ou à la couche de protocole ne suffit pas pour satisfaire à cette exigence, par exemple, signer une `<samlp:Response>` qui contient la ou les assertions ou une connexion TLS.

– `<AssertionConsumerService>` [Un ou plusieurs]

Un ou plusieurs éléments qui décrivent les points d'extrémité indexés qui prennent en charge les profils du protocole de demande d'authentification définis dans la présente Recommandation. Tout fournisseur de service accepte au moins un tel point d'extrémité, par définition.

– `<AttributeConsumingService>` [Zéro, un ou plusieurs]

Zéro, un ou plusieurs éléments qui décrivent une application ou service fourni par le fournisseur de service qui demande ou désire l'utilisation des attributs SAML.

Au plus un élément `<AttributeConsumingService>` peut avoir l'attribut `isDefault` mis à `vrai`. Il n'est permis à aucun des éléments inclus de contenir un attribut `isDefault` mis à `vrai`.

Le fragment de schéma suivant définit l'élément `<SPSSODescriptor>` et son type complexe **SPSSODescriptorType**:

```
<element name="SPSSODescriptor" type="md:SPSSODescriptorType"/>
<complexType name="SPSSODescriptorType">
       <complexContent>
              <extension base="md:SSODescriptorType">
                     <sequence>
                            <element ref="md:AssertionConsumerService"
maxOccurs="unbounded"/>
                            <element ref="md:AttributeConsumingService"
minOccurs="0" maxOccurs="unbounded"/>
                     </sequence>
                     <attribute name="AuthnRequestsSigned" type="boolean"
use="optional"/>
                     <attribute name="WantAssertionsSigned" type="boolean"
use="optional"/>
              </extension>
       </complexContent>
</complexType>
<element name="AssertionConsumerService" type="md:IndexedEndpointType"/>
```

9.1.4.4.1 Élément <AttributeConsumingService>

L'élément `<AttributeConsumingService>` définit un service particulier offert par le fournisseur de service en termes d'attributs que le service demande ou désire. Son type complexe **AttributeConsumingServiceType** contient les éléments et attributs suivants:

– `index` [Exigé]

Attribut exigé qui alloue une valeur d'entier unique à l'élément de sorte qu'il puisse être référencé dans un message de protocole.

– `isDefault` [Facultatif]

Identifie le service pris en charge par défaut par le fournisseur de service. Utile si le service spécifique n'est pas autrement indiqué par le contexte d'application. S'il est omis, la valeur est supposée être `faux`.

– `<ServiceName>` [Un ou plusieurs]

  Un ou plusieurs noms qualifiés en langage pour le service.

– `<ServiceDescription>` [Zéro, un ou plusieurs]

  Zéro, une ou plusieurs chaînes qualifiées en langage qui décrivent le service.

– `<RequestedAttribute>` [Un ou plusieurs]

  Un ou plusieurs éléments spécifiant les attributs exigés ou désirés par ce service.

Le fragment de schéma suivant définit l'élément `<AttributeConsumingService>` et son type complexe **AttributeConsumingServiceType**:

```
<element name="AttributeConsumingService"
type="md:AttributeConsumingServiceType"/>
<complexType name="AttributeConsumingServiceType">
      <sequence>
            <element ref="md:ServiceName" maxOccurs="unbounded"/>
            <element ref="md:ServiceDescription" minOccurs="0"
maxOccurs="unbounded"/>
            <element ref="md:RequestedAttribute" maxOccurs="unbounded"/>
      </sequence>
      <attribute name="index" type="unsignedShort" use="required"/>
      <attribute name="isDefault" type="boolean" use="optional"/>
</complexType>
<element name="ServiceName" type="md:localizedNameType"/>
<element name="ServiceDescription" type="md:localizedNameType"/>
```

9.1.4.4.2 Élément <RequestedAttribute>

L'élément `<RequestedAttribute>` spécifie l'intérêt d'un fournisseur de service pour un attribut SAML spécifique, incluant facultativement des valeurs spécifiques. Son type complexe **RequestedAttributeType** étend **saml:AttributeType** avec l'attribut suivant:

– `isRequired` [Facultatif]

  Attribut XML facultatif qui indique si le service exige l'attribut SAML correspondant pour fonctionner (par opposition à simplement trouver un attribut utile ou désirable).

  Si des éléments `<saml:AttributeValue>` spécifiques sont inclus, seules les valeurs qui correspondent sont pertinentes pour ce service.

  Le fragment de schéma suivant définit l'élément `<RequestedAttribute>` et son type complexe **RequestedAttributeType**:

```
<element name="RequestedAttribute" type="md:RequestedAttributeType"/>
<complexType name="RequestedAttributeType">
  <complexContent>
        <extension base="saml:AttributeType">
              <attribute name="isRequired" type="boolean"
use="optional"/>
        </extension>
  </complexContent>
</complexType>
```

9.1.4.5 Élément <AuthnAuthorityDescriptor>

L'élément `<AuthnAuthorityDescriptor>` étend **RoleDescriptorType** avec un contenu qui reflète les profils spécifiques des autorités d'authentification, autorités SAML qui répondent aux messages `<samlp:AuthnQuery>`. Son type complexe **AuthnAuthorityDescriptorType** contient les éléments supplémentaires suivants:

– `<AuthnQueryService>` [Un ou plusieurs]

  Un ou plusieurs éléments du type **EndpointType** qui décrivent les points d'extrémité qui prennent en charge le profil du protocole d'interrogation d'authentification défini au § 12. Toutes les autorités d'authentification acceptent au moins un tel point d'extrémité, par définition.

– `<AssertionIDRequestService>` [Zéro, un ou plusieurs]

  Zéro, un ou plusieurs éléments du type **EndpointType** qui décrivent les points d'extrémité qui prennent en charge le profil du protocole de demande d'assertion défini au § 12 ou la liaison d'URI spéciale pour la demande d'assertion définie au § 10.

– `<NameIDFormat>` [Zéro, un ou plusieurs]

Zéro, un ou plusieurs éléments du type **anyURI** qui énumèrent les formats d'identifiant de nom pris en charge par cette autorité (voir au § 8.7.3 les valeurs possibles de cet élément).

Le fragment de schéma suivant définit l'élément `<AuthnAuthorityDescriptor>` et son type complexe **AuthnAuthorityDescriptorType**:

```
<element name="AuthnAuthorityDescriptor"
type="md:AuthnAuthorityDescriptorType"/>
<complexType name="AuthnAuthorityDescriptorType">
       <complexContent>
              <extension base="md:RoleDescriptorType">
                     <sequence>
                            <element ref="md:AuthnQueryService"
maxOccurs="unbounded"/>
                            <element ref="md:AssertionIDRequestService"
minOccurs="0" maxOccurs="unbounded"/>
                            <element ref="md:NameIDFormat" minOccurs="0"
maxOccurs="unbounded"/>
                     </sequence>
              </extension>
       </complexContent>
</complexType>
<element name="AuthnQueryService" type="md:EndpointType"/>
```

9.1.4.6 Élément <PDPDescriptor>

L'élément `<PDPDescriptor>` étend **RoleDescriptorType** avec un contenu qui reflète les profils spécifiques des points de décision de politique, des autorités SAML qui répondent aux messages `<samlp:AuthzDecisionQuery>`. Son type complexe **PDPDescriptorType** contient l'élément supplémentaire suivant:

– `<AuthzService>` [Un ou plusieurs]

Un ou plusieurs éléments du type **EndpointType** qui décrivent les points d'extrémité qui prennent en charge le profil du protocole d'interrogation de décision d'autorisation défini au § 12. Tous les points de décision de politique acceptent au moins un tel point d'extrémité, par définition.

– `<AssertionIDRequestService>` [Zéro, un ou plusieurs]

Zéro, un ou plusieurs éléments du type **EndpointType** qui décrivent les points d'extrémité qui prennent en charge le profil du protocole de demande d'assertion défini au § 12 ou la liaison d'URI spéciale pour la demande d'assertion définie au § 10.

NOTE (informative) – PE33 (voir OASIS PE:2006) suggère de remplacer protocole de demande d'assertion par interrogation/demande d'assertion.

– `<NameIDFormat>` [Zéro, un ou plusieurs]

Zéro, un ou plusieurs éléments du type **anyURI** qui énumèrent les formats d'identifiant de nom pris en charge par cette autorité (voir au § 8.7.3 des valeurs possibles de cet élément).

Le fragment de schéma suivant définit l'élément `<PDPDescriptor>` et son type complexe **PDPDescriptorType**:

```
<element name="PDPDescriptor" type="md:PDPDescriptorType"/>
<complexType name="PDPDescriptorType">
       <complexContent>
              <extension base="md:RoleDescriptorType">
                     <sequence>
                            <element ref="md:AuthzService"
maxOccurs="unbounded"/>
                            <element ref="md:AssertionIDRequestService"
minOccurs="0" maxOccurs="unbounded"/>
                            <element ref="md:NameIDFormat" minOccurs="0"
maxOccurs="unbounded"/>
                     </sequence>
              </extension>
       </complexContent>
</complexType>
<element name="AuthzService" type="md:EndpointType"/>
```

9.1.4.7 Élément <AttributeAuthorityDescriptor>

L'élément `<AttributeAuthorityDescriptor>` étend **RoleDescriptorType** avec un contenu qui reflète les profils spécifiques des autorités d'attribut, les autorités SAML qui répondent aux messages `<samlp:AttributeQuery>`. Son type complexe **AttributeAuthorityDescriptorType** contient les éléments supplémentaires suivants:

– `<AttributeService>` [Un ou plusieurs]

Un ou plusieurs éléments du type **EndpointType** qui décrivent les points d'extrémité qui prennent en charge le profil du protocole d'interrogation d'attribut défini au § 12. Toutes les autorités d'attribut prennent en charge au moins un tel point d'extrémité, par définition.

– `<AssertionIDRequestService>` [Zéro, un ou plusieurs]

Zéro, un ou plusieurs éléments du type **EndpointType** qui décrivent les points d'extrémité qui prennent en charge le profil du protocole de demande d'assertion défini au § 12 ou la liaison d'URI spéciale pour la demande d'assertion définie au § 10.

NOTE (informative) – PE33 (voir OASIS PE:2006) suggère de remplacer protocole de demande d'assertion par interrogation/demande d'assertion.

– `<NameIDFormat>` [Zéro, un ou plusieurs]

Zéro, un ou plusieurs éléments du type **anyURI** qui énumèrent les formats d'identifiant de nom pris en charge par cette autorité (voir au § 8.7.3 les valeurs possibles de cet élément).

– `<AttributeProfile>` [Zéro, un ou plusieurs]

Zéro, un ou plusieurs éléments du type **anyURI** qui énumèrent les profils d'attribut pris en charge par cette autorité (voir au § 8.7.3 les valeurs possibles de cet élément).

– `<saml:Attribute>` [Zéro, un ou plusieurs]

Zéro, un ou plusieurs éléments qui identifient les attributs SAML pris en charge par l'autorité. Des valeurs spécifiques peuvent facultativement être incluses, indiquant que seules certaines valeurs permises par la définition de l'attribut sont prises en charge.

Le fragment de schéma suivant définit l'élément `<AttributeAuthorityDescriptor>` et son type complexe **AttributeAuthorityDescriptorType**:

```
<element name="AttributeAuthorityDescriptor"
type="md:AttributeAuthorityDescriptorType"/>
<complexType name="AttributeAuthorityDescriptorType">
       <complexContent>
              <extension base="md:RoleDescriptorType">
                     <sequence>
                            <element ref="md:AttributeService"
maxOccurs="unbounded"/>
                            <element ref="md:AssertionIDRequestService"
minOccurs="0" maxOccurs="unbounded"/>
                            <element ref="md:NameIDFormat" minOccurs="0"
maxOccurs="unbounded"/>
                            <element ref="md:AttributeProfile" minOccurs="0"
maxOccurs="unbounded"/>
                            <element ref="saml:Attribute" minOccurs="0"
maxOccurs="unbounded"/>
                     </sequence>
              </extension>
       </complexContent>
</complexType>
<element name="AttributeService" type="md:EndpointType"/>
```

9.1.5 Élément <AffiliationDescriptor>

L'élément `<AffiliationDescriptor>` est une solution de remplacement pour la séquence des descripteurs de rôle qui est utilisée lorsqu'un `<EntityDescriptor>` décrit une affiliation d'entités SAML (normalement des fournisseurs de service) plutôt qu'une seule entité. L'élément `<AffiliationDescriptor>` fournit un résumé des entités individuelles qui constituent l'affiliation ainsi que des informations générales sur l'affiliation elle-même. Son type complexe **AffiliationDescriptorType** contient les éléments et attributs suivants:

– `affiliationOwnerID` [Exigé]

  Spécifie l'identifiant unique de l'entité responsable de l'affiliation. Le propriétaire n'est pas supposé être un membre de l'affiliation; s'il est membre, son identifiant doit aussi apparaître dans un élément `<AffiliateMember>`.

– `ID` [Facultatif]

  Identifiant de l'élément unique pour le document, normalement utilisé comme point de référence à la signature.

– `validUntil` [Facultatif]

  Attribut facultatif qui indique l'heure d'expiration des métadonnées contenues dans l'élément et tout élément contenu.

– `cacheDuration` [Facultatif]

  Attribut facultatif qui indique la durée maximale pendant laquelle un consommateur devrait conserver en mémoire cache les métadonnées contenues dans l'élément et tout élément contenu.

– `<ds:Signature>` [Facultatif]

  Signature XML qui authentifie l'élément contenant et son contenu (voir au § 8).

– `<Extensions>` [Facultatif]

  Contient des extensions de métadonnées facultatives qui sont convenues entre un éditeur et un consommateur de métadonnées. Les éléments d'extension doivent être qualifiés en espace de nom par un espace de nom non défini par SAML.

– `<AffiliateMember>` [Un ou plusieurs]

  Un ou plusieurs éléments énumérant les membres de l'affiliation en spécifiant l'identifiant unique de chaque membre (voir aussi au § 8.7.3.6).

– `<KeyDescriptor>` [Zéro, un ou plusieurs]

  Séquence facultative d'éléments qui fournit des informations sur les clés cryptographiques qu'utilise l'affiliation comme un tout, distinctes des clés utilisées par les membres individuels de l'affiliation, qui sont publiées dans les métadonnées pour ces entités.

Des attributs arbitraires qualifiés en espace de nom provenant d'espaces de nom non définis par SAML peuvent aussi être inclus.

Le fragment de schéma suivant définit l'élément `<AffiliationDescriptor>` et son type complexe **AffiliationDescriptorType**:

```
<element name="AffiliationDescriptor" type="md:AffiliationDescriptorType"/>
<complexType name="AffiliationDescriptorType">
       <sequence>
              <element ref="ds:Signature" minOccurs="0"/>
              <element ref="md:Extensions" minOccurs="0"/>
              <element ref="md:AffiliateMember" maxOccurs="unbounded"/>
              <element ref="md:KeyDescriptor" minOccurs="0"
maxOccurs="unbounded"/>
       </sequence>
       <attribute name="affiliationOwnerID" type="md:entityIDType"
use="required"/>
       <attribute name="validUntil" type="dateTime" use="optional"/>
       <attribute name="cacheDuration" type="duration" use="optional"/>
       <attribute name="ID" type="ID" use="optional"/>
       <anyAttribute namespace="##other" processContents="lax"/>
</complexType>
<element name="AffiliateMember" type="md:entityIDType"/>
```

9.1.6 Exemples

Exemple de métadonnées pour une entité système SAML jouant le rôle d'un fournisseur d'identité et d'une autorité d'attribut. Une signature figure comme marqueur de position, sans contenu réel.

```
<EntityDescriptor xmlns="urn:oasis:names:tc:SAML:2.0:metadata"
       xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion"
       xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
```

```
      entityID="https://IdentityProvider.com/SAML">
      <ds:Signature>...</ds:Signature>
    <IDPSSODescriptor WantAuthnRequestsSigned="true"

      protocolSupportEnumeration="urn:oasis:names:tc:SAML:2.0:protocol">
       <KeyDescriptor use="signing">
           <ds:KeyInfo>
               <ds:KeyName>IdentityProvider.com SSO Key</ds:KeyName>
           </ds:KeyInfo>
       </KeyDescriptor>
       <ArtifactResolutionService isDefault="true" index="0"
                   Binding="urn:oasis:names:tc:SAML:2.0:bindings:SOAP"
                   Location="https://IdentityProvider.com/SAML/Artifact"/>
       <SingleLogoutService
                   Binding="urn:oasis:names:tc:SAML:2.0:bindings:SOAP"
                   Location="https://IdentityProvider.com/SAML/SLO/SOAP"/>
       <SingleLogoutService
                   Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-
Redirect"

      Location="https://IdentityProvider.com/SAML/SLO/Browser"

      ResponseLocation="https://IdentityProvider.com/SAML/SLO/Response"/>
       <NameIDFormat>
            urn:oasis:names:tc:SAML:1.1:nameid-format:X509SubjectName
       </NameIDFormat>
       <NameIDFormat>
            urn:oasis:names:tc:SAML:2.0:nameid-format:persistent
       </NameIDFormat>
       <NameIDFormat>
            urn:oasis:names:tc:SAML:2.0:nameid-format:transient
       </NameIDFormat>
       <SingleSignOnService
                   Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-
Redirect"

      Location="https://IdentityProvider.com/SAML/SSO/Browser"/>
       <SingleSignOnService
                   Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-
POST"

      Location="https://IdentityProvider.com/SAML/SSO/Browser"/>
       <saml:Attribute
                   NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-
format:uri"
                   Name="urn:oid:1.3.6.1.4.1.5923.1.1.1.6"
                   FriendlyName="eduPersonPrincipalName">
       </saml:Attribute>
       <saml:Attribute
                   NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-
format:uri"
                   Name="urn:oid:1.3.6.1.4.1.5923.1.1.1.1"
                   FriendlyName="eduPersonAffiliation">
           <saml:AttributeValue>member</saml:AttributeValue>
           <saml:AttributeValue>student</saml:AttributeValue>
           <saml:AttributeValue>faculty</saml:AttributeValue>
           <saml:AttributeValue>employee</saml:AttributeValue>
           <saml:AttributeValue>staff</saml:AttributeValue>
       </saml:Attribute>
    </IDPSSODescriptor>
    <AttributeAuthorityDescriptor

      protocolSupportEnumeration="urn:oasis:names:tc:SAML:2.0:protocol">
       <KeyDescriptor use="signing">
           <ds:KeyInfo>
               <ds:KeyName>IdentityProvider.com AA Key</ds:KeyName>
           </ds:KeyInfo>
       </KeyDescriptor>
       <AttributeService
                   Binding="urn:oasis:names:tc:SAML:2.0:bindings:SOAP"
                   Location="https://IdentityProvider.com/SAML/AA/SOAP"/>
```

```
        <AssertionIDRequestService
                    Binding="urn:oasis:names:tc:SAML:2.0:bindings:URI"
                    Location="https://IdentityProvider.com/SAML/AA/URI"/>
        <NameIDFormat>
             urn:oasis:names:tc:SAML:1.1:nameid-format:X509SubjectName
        </NameIDFormat>
        <NameIDFormat>
             urn:oasis:names:tc:SAML:2.0:nameid-format:persistent
        </NameIDFormat>
        <NameIDFormat>
             urn:oasis:names:tc:SAML:2.0:nameid-format:transient
        </NameIDFormat>
        <saml:Attribute
                    NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-
format:uri"
                    Name="urn:oid:1.3.6.1.4.1.5923.1.1.1.6"
                    FriendlyName="eduPersonPrincipalName">
        </saml:Attribute>
        <saml:Attribute
                    NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-
format:uri"
                    Name="urn:oid:1.3.6.1.4.1.5923.1.1.1.1"
                    FriendlyName="eduPersonAffiliation">
            <saml:AttributeValue>member</saml:AttributeValue>
            <saml:AttributeValue>student</saml:AttributeValue>
            <saml:AttributeValue>faculty</saml:AttributeValue>
            <saml:AttributeValue>employee</saml:AttributeValue>
            <saml:AttributeValue>staff</saml:AttributeValue>
        </saml:Attribute>
    </AttributeAuthorityDescriptor>
    <Organization>
        <OrganizationName xml:lang="en">Identity Providers R
US</OrganizationName>
        <OrganizationDisplayName xml:lang="en">
             Identity Providers R US, a Division of Lerxst Corp.
        </OrganizationDisplayName>
        <OrganizationURL
xml:lang="en">https://IdentityProvider.com</OrganizationURL>
    </Organization>
</EntityDescriptor>
```

Exemple de métadonnées pour une entité système SAML jouant le rôle de fournisseur de service. Une signature figure comme marqueur de position, sans contenu réel. Pour les besoins de l'exemple, le service est un de ceux qui n'exigent pas des utilisateurs qu'ils s'identifient de façon univoque, mais plutôt autorisent l'accès sur la base d'un attribut de rôle.

```
<EntityDescriptor xmlns="urn:oasis:names:tc:SAML:2.0:metadata"
       xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion"
       xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
       entityID="https://ServiceProvider.com/SAML">
       <ds:Signature>...</ds:Signature>
    <SPSSODescriptor AuthnRequestsSigned="true"

       protocolSupportEnumeration="urn:oasis:names:tc:SAML:2.0:protocol">
        <KeyDescriptor use="signing">
            <ds:KeyInfo>
                <ds:KeyName>ServiceProvider.com SSO Key</ds:KeyName>
            </ds:KeyInfo>
        </KeyDescriptor>
        <KeyDescriptor use="encryption">
            <ds:KeyInfo>
                <ds:KeyName>ServiceProvider.com Encrypt Key</ds:KeyName>
            </ds:KeyInfo>
            <EncryptionMethod
       Algorithm="http://www.w3.org/2001/04/xmlenc#rsa-1_5"/>
        </KeyDescriptor>
        <SingleLogoutService
                    Binding="urn:oasis:names:tc:SAML:2.0:bindings:SOAP"
                    Location="https://ServiceProvider.com/SAML/SLO/SOAP"/>
        <SingleLogoutService
                    Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-
Redirect"
```

```
                    Location="https://ServiceProvider.com/SAML/SLO/Browser"

       ResponseLocation="https://ServiceProvider.com/SAML/SLO/Response"/>
         <NameIDFormat>
              urn:oasis:names:tc:SAML:2.0:nameid-format:transient
         </NameIDFormat>
         <AssertionConsumerService isDefault="true" index="0"
                    Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-
Artifact"

       Location="https://ServiceProvider.com/SAML/SSO/Artifact"/>
         <AssertionConsumerService index="1"
                    Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-
POST"
                    Location="https://ServiceProvider.com/SAML/SSO/POST"/>
         <AttributeConsumingService index="0">
             <ServiceName xml:lang="en">Academic Journals R US</ServiceName>
             <RequestedAttribute
                          NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-
format:uri"
                          Name="urn:oid:1.3.6.1.4.1.5923.1.1.1.7"
                          FriendlyName="eduPersonEntitlement">
                 <saml:AttributeValue>
                     https://ServiceProvider.com/entitlements/123456789
                 </saml:AttributeValue>
             </RequestedAttribute>
         </AttributeConsumingService>
     </SPSSODescriptor>
     <Organization>
         <OrganizationName xml:lang="en">Academic Journals R
US</OrganizationName>
         <OrganizationDisplayName xml:lang="en">
                Academic Journals R US, a Division of Dirk Corp.
         </OrganizationDisplayName>
         <OrganizationURL
xml:lang="en">https://ServiceProvider.com</OrganizationURL>
     </Organization>
</EntityDescriptor>
```

9.2 Traitement de signature

Divers éléments d'une instance de métadonnées peuvent être signés numériquement (comme indiqué par l'inclusion d'un élément `<ds:Signature>`), avec les avantages suivants:

9.2.1 Intégrité des métadonnées

Authentification des métadonnées par un signataire de confiance.

Une signature numérique n'est pas toujours exigée, par exemple si le consommateur d'assertions obtient les informations directement de l'entité éditrice (sans intermédiaire) par un canal sécurisé, l'entité s'étant authentifiée auprès du consommateur d'assertions par un moyen autre qu'une signature numérique.

De nombreuses techniques différentes sont disponibles pour l'authentification "directe" et l'établissement d'un canal sécurisé entre deux parties. La liste inclut TLS, HMAC, des mécanismes fondés sur un mot de passe, etc. De plus, les exigences de sécurité applicables dépendent des applications communicantes.

De surcroît, les éléments peuvent hériter de signatures en incluant des éléments parents qui sont eux-mêmes signés.

En l'absence d'un tel contexte, il est recommandé qu'au moins l'élément racine d'une instance de métadonnées soit signé.

9.2.2 Profil de signature XML

La spécification Signature XML du W3C appelle une syntaxe générale XML souple et avec de nombreux choix pour les données de signature. La présente section détaille les contraintes qui pèsent sur ces facilités afin que les processeurs de métadonnées n'aient pas à traiter du processus de signature XML dans toute sa généralité. Cette utilisation s'appuie spécifiquement sur les attributs de type **xs:ID** facultativement présents sur les éléments auxquels les signatures peuvent s'appliquer. Ces attributs sont collectivement désignés dans cette section sous le nom d'attributs identifiants.

1) Formats et algorithmes de signature

Signature XML a trois façons de rapporter une signature à un document: enveloppante, enveloppée, et détachée.

Les métadonnées SAML doivent utiliser des signatures enveloppées lors de la signature des éléments définis dans la présente Recommandation. Les processeurs SAML devraient prendre en charge l'utilisation de la signature RSA et la vérification des opérations de clés publiques conformément à l'algorithme identifié par http://www.w3.org/2000/09/xmldsig#rsa-sha1.

2) Références

Les éléments de métadonnées signés doivent fournir une valeur pour l'attribut identifiant sur l'élément signé. L'élément peut être ou non l'élément racine du document XML réel qui contient l'élément de métadonnées signé.

Les signatures doivent contenir un seul `<ds:Reference>` contenant une référence d'URI à la valeur d'attribut identifiant de l'élément de métadonnées à signer. Par exemple, si la valeur d'attribut identifiant est "foo", l'attribut d'URI dans l'élément `<ds:Reference>` doit alors être "#foo".

En conséquence, la signature d'un élément de métadonnées doit s'appliquer au contenu de l'élément signé et de tout élément fils qu'il contient.

3) Méthode de canonisation

Les implémentations SAML devraient utiliser la canonisation exclusive, avec ou sans commentaire, à la fois dans l'élément `<ds:CanonicalizationMethod>` de `<ds:SignedInfo>`, et comme un algorithme `<ds:Transform>`. L'utilisation de la canonisation exclusive assure que les signatures créées sur des métadonnées SAML enchassées dans un contexte XML peuvent être vérifiées indépendamment de ce contexte.

4) Transformations

Les signatures dans les métadonnées SAML ne devraient pas contenir de transformations autres que les transformations de signature enveloppée (avec l'identifiant http://www.w3.org/2000/09/xmldsig#enveloped-signature) ou les transformations canoniques exclusives (avec l'identifiant http://www.w3.org/2001/10/xml-exc-c14n# ou http://www.w3.org/2001/10/xml-exc-c14n#WithComments).

Les vérificateurs de signatures peuvent rejeter les signatures qui contiennent d'autres algorithmes de transformation comme non valides. S'ils ne le font pas, les vérificateurs doivent s'assurer qu'aucun contenu de l'élément de métadonnées signé n'est exclu de la signature. Ceci peut se faire en établissant une convention hors bande selon laquelle les transformations sont acceptables, ou en appliquant les transformations manuellement au contenu et en revérifiant que le résultat consiste bien en les mêmes métadonnées SAML.

5) KeyInfo

Signature XML du W3C définit l'usage de l'élément `<ds:KeyInfo>`. SAML n'exige pas l'utilisation de `<ds:KeyInfo>` ni n'impose de restrictions à son utilisation. Donc, `<ds:KeyInfo>` peut être absent.

9.3 Publication et résolution des métadonnées

Dans la présente Recommandation, deux mécanismes sont fournis pour qu'une entité publie (et pour qu'un consommateur résolve la localisation) des documents de métadonnées: via une "localisation bien connue" en déréférençant directement l'identifiant unique de l'entité (un URI diversement référencé comme un *entityID* ou *providerID*), ou indirectement en publiant la localisation des métadonnées dans le DNS. D'autres mécanismes hors bande sont bien sûr aussi permis. Un consommateur qui accepte les deux approches doit essayer d'abord la résolution via DNS avant d'utiliser le mécanisme de "localisation bien connue".

Lorsque la restitution exige un transport réseau du document, le transport devrait être protégé par des mécanismes fournissant l'authentification du serveur et la protection de l'intégrité. Par exemple, la résolution fondée sur HTTP devrait être protégée par TLS comme défini dans la RFC 2246 de l'IETF amendée par la RFC 3546 de l'IETF.

Divers mécanismes sont décrits dans la présente section pour aider à établir la confiance dans l'exactitude et la légitimité des métadonnées, y compris l'utilisation des signatures XML, l'authentification de serveur TLS, et les signatures DNS. Quels que soient le ou les mécanismes utilisés, les consommateurs d'assertions devraient avoir des moyens d'établir la confiance dans les informations de métadonnées avant de s'appuyer sur elles.

9.3.1 Publication et résolution via une localisation bien connue

Les paragraphes suivants décrivent la publication et la résolution des métadonnées au moyen d'une localisation bien connue.

9.3.1.1 Publication

Les entités peuvent publier leur documents de métadonnées à une localisation bien connue en plaçant le document à la localisation notée par son identifiant unique, qui doit être sous la forme d'un URL (plutôt qu'un URN). Il est fortement recommandé que les URL https soient utilisés à cette fin. Un mécanisme directionnel pris en charge par le schéma d'URL (tel qu'un HTTP 1.1 302 redirect) peut être utilisé si le document n'est pas placé directement à la localisation. Si le protocole de publication permet l'identification fondée sur MIME des types de contenu, le type de contenu de l'instance de métadonnées doit être `application/samlmetadata+xml`.

Le document XML fourni à la location bien connue doit seulement décrire les métadonnées pour l'entité représentée par l'identifiant unique (c'est-à-dire que l'élément racine doit être un `<EntityDescriptor>` avec une entityID correspondant à la localisation). Si d'autres entités doivent être décrites, l'élément `<AdditionalMetadataLocation>` doit être utilisé. Et donc, l'élément `<EntitiesDescriptor>` ne doit pas être utilisé dans des documents publiés en utilisant ce mécanisme, car un groupe d'entités n'est pas défini par un tel identifiant.

9.3.1.2 Résolution

Si l'identifiant unique d'une entité est un URL, les consommateurs de métadonnées peuvent essayer de résoudre un identifiant unique d'entité directement, d'une façon spécifique au schéma, en déréférençant l'identifiant.

9.3.2 Publication et résolution via DNS

Pour améliorer l'accessibilité des documents de métadonnées et fournir des correspondances supplémentaires entre un identifiant unique d'entité et la localisation des métadonnées, les entités peuvent publier leurs localisations de document de métadonnées dans une zone de leur DNS correspondant, comme défini dans la RFC 1034 de l'IETF. L'identifiant unique de l'entité (un URI) est utilisé comme entrée du processus. Comme les URI sont des identifiants souples, les méthodes de publication de localisation et le processus de résolution sont déterminés par le schéma d'URI et le nom pleinement qualifié. Les localisations d'URI pour les métadonnées peuvent ensuite être déduites par des interrogations du registre des ressources (RR) NAPTR comme défini dans la RFC 2914 et la RFC 3403 de l'IETF.

Il est recommandé que les entités publient leurs enregistrements de ressources dans des fichiers de zone signée en utilisant la RFC 2535 de l'IETF, de sorte que les consommateurs d'assertions puissent établir la validité des localisations publiées et de l'autorité de la zone, et l'intégrité de la réponse DNS. Si les signatures de zone DNS sont présentes, les consommateurs d'assertions doivent valider correctement la signature.

9.3.2.1 Publication

La présente Recommandation utilise l'enregistrement de ressource NAPTR décrit dans la RFC 2915 et la RFC 3403 de l'IETF. Il est recommandé de se familiariser avec ces documents.

Le système dynamique de recherche de délégation (DDDS, *dynamic delegation discovery system*) est un système générique pour la restitution d'informations sur la base d'une chaîne d'entrées spécifique d'une application et l'application de règles bien connues pour transformer cette chaîne jusqu'à atteindre une condition finale exigeant une recherche dans une base de données définie spécifique d'une application ou la résolution d'un URL sur la base des règles définies par l'application. DDDS définit un type spécifique d'enregistrement de ressources DNS, les enregistrements NAPTR, pour le stockage d'informations nécessaires pour appliquer les règles DDDS dans le DNS.

Les entités peuvent publier des URL séparés lorsque plusieurs documents de métadonnées doivent être distribués, ou lorsque différents documents de métadonnées sont nécessaires du fait de relations de confiance multiples qui exigent du matériel de chiffrement séparé, ou lorsque les interfaces de service exigent des déclarations de métadonnées séparées. Cela peut se faire en utilisant l'élément facultatif `<AdditionalMetadataLocation>`, ou par la facilité "regexp" et plusieurs champs de définition de service dans l'enregistrement de ressources NAPTR lui-même.

Si le protocole de publication permet l'identification fondée sur MIME des types de contenu, le type de contenu de l'instance de métadonnées doit être `application/samlmetadata+xml`.

Si l'identifiant unique de l'entité est un URN, la publication de la localisation de métadonnées correspondante se passe comme spécifié dans la RFC 3404 de l'IETF. Autrement, la résolution de la localisation des métadonnées se passe comme spécifié ci-dessous.

Ce qui suit est le profil spécifique d'application de DDDS pour la résolution de métadonnées SAML:

1) Première règle bien connue

La "première règle bien connue" pour traiter la résolution de métadonnées SAML est de faire l'analyse de l'identifiant unique de l'entité et d'extraire le nom de domaine pleinement qualifié (sous-expression 3).

2) Le champ d'ordre

Le champ d'ordre indique l'ordre de traitement de chaque enregistrement de ressource NAPTR retourné. Les éditeurs peuvent fournir plusieurs enregistrements de ressource NAPTR qui doivent être traités par l'application de résolution dans l'ordre indiqué par ce champ.

3) Le champ préference

Pour les enregistrements terminaux de ressource NAPTR, l'éditeur exprime l'ordre préféré d'utilisation de l'application de résolution. L'application de résolution peut ignorer cet ordre, dans les cas où la valeur de champ de service ne correspond pas aux exigences de l'équipement de résolution (par exemple, l'enregistrement de ressource retourne un protocole que l'application ne prend pas en charge).

4) Le champ fanion

La résolution de métadonnées SAML utilise deux fois le fanion "U", qui est terminal, et la valeur nulle (qui implique que des enregistrements de ressource supplémentaires sont à traiter). Le fanion "U" indique que le résultat de la règle est un URI.

5) Le champ service

Le champ service spécifique de SAML, comme décrit dans le BNF suivant, déclare les modes selon lesquels la ou les instances de document seront rendues disponibles:

```
servicefield = 1("PID2U" / "NID2U") "+" proto [*(":" class) *(":"
servicetype)]
proto = 1("https" / "uddi")
class = 1[ "entity" / "entitygroup" )
servicetype = 1(si / "spsso" / "idpsso" / "authn" / "authnauth" / "pdp" /
"attrauth" / alphanum )
si = "si" [":" alphanum] [":endpoint"]
alphanum = 1*32(ALPHA / DIGIT)
```

où:

- `servicefield` PID2U résout un identifiant unique d'entité en URL de métadonnées.
- `servicefield` NID2U résout un `<NameID>` de principal en URL de métadonnées.
- `proto` décrit le protocole de restitution (`https` ou `uddi`). Dans le cas de UDDI, l'URL sera un URL http(s) qui référence un document WSDL.
- `class` identifie si le document de métadonnées référencé décrit une seule entité, ou plusieurs. Dans ce dernier cas, le document référence doit contenir l'entité définie par l'identifiant unique d'origine comme un membre d'un groupe d'entités au sein du document lui-même tel qu'un `<AffiliationDescriptor>` ou `<EntitiesDescriptor>`.
- `servicetype` permet à une entité de publier des métadonnées pour des rôles et services distincts comme documents séparés. Les mécanismes de résolution qui rencontrent plusieurs déclarations `servicetype` déréférenceront l'URI approprié, en fonction du service qui est requis pour une opération (par exemple, une entité fonctionnant à la fois comme un fournisseur d'identité et comme un fournisseur de service peut publier des métadonnées pour chaque rôle à des localisations différentes). Le type de service authn représente un point d'extrémité `<SingleSignOnService>`.
- `si` (avec un composant de point d'extrémité facultatif) permet à l'éditeur de publier directement les métadonnées pour une instance de service, ou en articulant un point d'extrémité SOAP (en utilisant un point d'extrémité `endpoint`).

Par exemple:

- `PID2U+https:entity` – représente le document de métadonnées complet de l'entité disponible via le protocole https.
- `PID2U+uddi:entity:si:foo` – représente la localisation de document WSDL qui décrit une instance de service "foo".

– `PID2U+https:entitygroup:idpsso` – représente les métadonnées pour un groupe d'entités agissant comme fournisseurs d'identité SSO, dont l'entité d'origine est membre.

– `NID2U+https:idp` – représente les métadonnées pour le fournisseur d'identité SSO d'un principal.

6) Les champs regex et replacement

Le résultat attendu après traitement de la chaîne d'entrées à travers le regex doit être un URL `https` valide ou une adresse de nœud UDDI (document WSDL).

9.3.2.2 Exemples de NAPTR

Le présent paragraphe donne des exemples d'URL et d'adresses de messagerie électronique qui peuvent être utilisés par des entités qui acceptent NAPTR (voir la RFC 2915 de l'IETF).

a) Exemples NAPTR de métadonnées d'entité

Les entitiés publient les URL de métadonnées de la manière suivante:

```
$ORIGIN provider.biz

;; order pref f service regexp or replacement

IN NAPTR 100 10 "U" PID2U+https:entity
 "!^.*$!https://host.provider.biz/some/directory/trust.xml!" ""
IN NAPTR 110 10 "U" PID2U+https: entity:trust
 "!^.*!https://foo.provider.biz:1443/mdtrust.xml!" ""
IN NAPTR 125 10 "U" PID2U+https:"
IN NAPTR 110 10 "U" PID2U+uddi:entity
 "!^.*$!https://this.uddi.node.provider.biz/libmd.wsdl" ""
```

b) Exemples d'identifiant de nom

Un employeur de principal `example.int` fait fonctionner un fournisseur d'identité qui peut être utilisé par une société de fournitures de bureau pour authentifier les acheteurs autorisés. Le fournisseur prend une adresse de messagerie électronique d'utilisateur `buyer@example.int` comme entrée du processus de résolution, et analyse l'adresse de messagerie pour extraire le FQDN (`example.int`). L'employeur publie l'enregistrement NAPTR suivant dans le DNS `example.int`:

```
$ORIGIN example.int

IN NAPTR 100 10 "U" NID2U+https:authn
 "!^([^@]+)@(.*)$!https://serv.example.int:8000/cgi-bin/getmd?\1!" ""
IN NAPTR 100 10 "U" NID2U+https:idp
 "!^([^@]+)@(.*)$!https://auth.example.int/app/auth?\1" ""
```

9.3.2.3 Résolution

Lors de la résolution de métadonnées pour une entité via le DNS, l'identifiant unique de l'entité est utilisé comme l'entrée initiale du processus de résolution, plutôt que comme une localisation réelle. Procéder comme suit:

– si l'identifiant unique est un URN, poursuivre les étapes de résolution comme défini dans la RFC 3403 de l'IETF;

– autrement, analyser l'identifiant pour obtenir le nom de domaine pleinement qualifié;

– interroger itérativement le DNS sur les enregistrements de ressources NAPTR du domaine jusqu'à ce qu'un enregistrement de ressources terminales soit retourné;

– identifier quel enregistrement de ressource utiliser sur la base des champs de service, puis des champs d'ordre, puis des champs de préférence de l'ensemble des résultats;

– obtenir le ou les documents à la ou les localisations fournies comme demandé par l'application.

Pour initier la résolution de la localisation des informations de métadonnées, il sera nécessaire dans certains cas de décomposer l'identifiant unique de l'entité (exprimé comme un URI) en un ou plusieurs éléments atomiques.

L'expression régulière suivante devrait être utilisée lors de l'initialisation du processus de décomposition:

```
^([^:/?#]+:)?/*([^:/?#]*@)?(([^/?:#]*\.)*(([^/?#:\.]+)\.([^/?#:\.]+)))(:\d+)?([^?
#]*)(\?[^#]*)?(#.*)?$
     1            2           34            56             7             8       9
10        11
```

La sous-expression 3 doit avoir pour résultat un nom de domaine pleinement qualifié (FQDN, *fully-qualified domain name*), qui sera la base de la restitution des localisations de métadonnées à partir de cette zone.

A l'achèvement de l'analyse de l'identifiant, l'application effectue alors une interrogation DNS sur le domaine résultant (sous-expression 5) pour les enregistrements de ressources NAPTR; on devrait s'attendre à une ou plusieurs réponses. Les applications peuvent exclure de l'ensemble de résultats toute définition de service qui ne concerne pas les opérations de demande en cours.

Les applications de résolution doivent ensuite ordonner l'ensemble de résultats conformément au champ d'ordre, et peuvent ordonner l'ensemble de résultats sur la base de l'ensemble des préférences. Les systèmes de résolution ne sont pas obligés de suivre l'ordre du champ de préférences. Le ou les enregistrements de ressources NAPTR résultants sont traités de façon itérative (sur la base du fanion d'ordre) jusqu'à ce qu'un enregistrement de ressources NAPTR terminales soit atteint.

Le résultat sera un URL absolu, bien formé, qui sera ensuite utilisé pour restituer le document de métadonnées.

9.3.2.4 Mise en mémoire cache de localisation de métadonnées

La mise en mémoire cache de localisation de métadonnées ne doit pas excéder la durée de vie de la zone DNS à partir duquel la localisation a été déduite. Les systèmes de résolution doivent obtenir une copie récente de la localisation des métadonnées lorsque est atteinte l'expiration de la durée de vie de la zone.

Les éditeurs de documents de métadonnées devrait considérer avec attention la durée de vie de la zone lorsqu'ils font des changements de localisation de document de métadonnées. Si un tel changement de localisation doit survenir, l'éditeur doit garder le document dans les deux localisations, ancienne et nouvelle, jusqu'à ce que tous les systèmes de résolution conformes soient certains d'avoir la localisation mise à jour (par exemple, heure du changement de zone + durée de vie), ou fournir une réponse Redirect HTTP à l'ancienne localisation qui spécifie la nouvelle localisation.

9.3.3 Post-traitement de métadonnées

Les paragraphes qui suivent décrivent le post-traitement de métadonnées.

9.3.3.1 Mise en mémoire cache d'une instance de métadonnées

La mise en mémoire cache de document ne doit pas excéder l'attribut `validUntil` ou `cacheDuration` du ou des éléments sujets. Si les éléments de métadonnées ont des éléments parents qui contiennent des politiques de mise en mémoire cache, l'élément parent a la priorité.

Pour traiter de façon appropriée l'attribut `cacheDuration`, les consommateurs doivent retenir la date et l'heure à laquelle le document a été restitué.

Lorsqu'un document ou élément est arrivé à expiration, le consommateur doit récupérer une copie fraîche, qui peut exiger un rafraîchissement de la ou des localisations du document. Les consommateurs devraient traiter la mise en mémoire cache des documents conformément au § 13 de la RFC 2616 de l'IETF et peuvent demander la dernière date et heure modifiées au serveur HTTP. Les éditeurs devraient s'assurer d'un traitement acceptable de mise en mémoire cache, comme décrit au § 10.3.5 de la RFC 2616 de l'IETF (304 Non Modifié).

9.3.3.2 Traitement des renvois HTTPS

L'éditeur peut produire un HTTP Redirect (301 Déplacement permanent, 302 ou 307 Renvoi temporaire) comme défini dans la RFC 2616 de l'IETF, et les agents d'utilisateur doivent suivre l'URL spécifié dans la réponse Redirect. Les renvois devraient être du même protocole que la demande initiale.

9.3.3.3 Traitement des signatures XML et traitement général de confiance

Le traitement des métadonnées fournit plusieurs des mécanismes pour la négociation de sécurité à la fois pour les métadonnées elles-mêmes et pour la confiance accordée à l'entité décrite par de telles métadonnées:

- confiance déduite de la signature de la zone DNS à partir de laquelle l'URL de localisation des métadonnées a été résolu, assurant l'exactitude de la ou des localisation de documents de métadonnées;

– confiance déduite du traitement de la signature du document de métadonnées lui-même, assurant l'intégrité du document XML;

– confiance déduite de l'authentification de serveur TLS de l'URL de localisation des métadonnées, assurant l'identité de l'éditeur des métadonnées.

Le post-traitement du document de métadonnées doit inclure le traitement de signature au niveau du document XML et peut inclure un des deux autres processus. Précisément, le consommateur d'assertions peut choisir de faire confiance à toute autorité citée dans le processus de résolution et d'analyse. Les éditeurs de métadonnées doivent employer un mécanisme de protection de l'intégrité du document et peuvent employer un des deux autres profils de traitement pour établir la confiance envers le document de métadonnées, en fonction des politiques d'implémentation. Les considérations suivantes doivent être prises en compte:

1) Traitement des zones DNS signées

La vérification de la signature de zone DNS devrait être effectuée, si elle est présente, comme décrit dans la RFC 2535 de l'IETF.

2) Traitement des documents et fragments signés

Les documents de métadonnés publiés devraient être signés, comme décrit dans la présente Recommandation, soit par un certificat produit au sujet du document, soit auprès d'une autre partie de confiance. Les éditeurs peuvent considérer les signatures des autres parties comme un moyen de convoyer la confiance.

Les consommateurs de métadonnées doivent valider les signatures, lorqu'elles sont présentes, sur le document de métadonnées comme décrit dans la présente Recommandation.

3) Traitement de l'authentification du serveur durant la restitution des métadonnées via TLS

Il est fortement recommandé que les éditeurs implémentent les URL TLS; et donc les consommateurs devraient considérer la confiance héritée du producteur du certificat TLS. Les URL de publication ne peuvent pas toujours être localisés dans le domaine du sujet du document de métadonnées; donc les consommateurs ne devraient pas présumer des certificats dont le sujet est l'entité en question, car il peut être hébergé par une autre partie de confiance.

Comme la base de cette confiance peut n'être pas disponible à l'égard d'un document placé en mémoire cache, d'autres mécanismes devraient être utilisés dans de telles circonstances.

10 Liaisons pour SAML

Le présent paragraphe spécifie les liaisons de protocole SAML à l'usage des assertions SAML et des messages de demande et réponses dans les protocoles et cadres de travail de communications.

Les mappages d'échanges de messages de demande-réponse SAML en échange de messages standards ou protocoles de communication sont appelés *liaisons de protocole* (ou simplement *liaisons*) SAML. Une instance de mappage d'échanges de messages de demande-réponse SAML en protocole de communication spécifique `<FOO>` est appelée une *liaison <FOO> pour SAML* ou une *liaison <FOO> SAML*.

Par exemple, une liaison SOAP SAML décrit comment les échanges de messages de demande et réponse SAML sont mappés en échanges de messages SOAP.

L'intention de la présente Recommandation est de spécifier un ensemble choisi de liaisons à un niveau de détail suffisant pour assurer que les logiciels conformes à SAML qui sont implémentés de façon indépendante peuvent interopérer lorsqu'ils utilisent les échanges de messages standards ou les protocoles de communication.

Sauf spécification contraire, une liaison devrait être comprise comme support de la transmission de tout message de protocole SAML déduit des types **samlp:RequestAbstractType** et **samlp:StatusResponseType**. De plus, lorsqu'une liaison se réfère aux "demandes et réponses SAML", elle devrait être comprise comme signifiant tout message de protocole dérivé de ces types.

La présente Recommandation utilise les conventions typographiques suivantes dans le texte: `<ns:Element>`, XMLAttribute, **Datatype**, OtherKeyword. Dans certains cas, des crochets angulaires sont utilisés pour indiquer des éléments non terminaux, plutôt que XML; l'intention ressortira clairement du contexte.

10.1 Lignes directrices pour spécifier des liaisons de protocole supplémentaires

La présente Recommandation définit un ensemble choisi de liaisons de protocole, mais d'autres seront peut-être développées à l'avenir. Le présent paragraphe propose des lignes directrices pour les tierces parties qui souhaitent spécifier des liaisons supplémentaires. Ci-après figure une liste de contrôle des questions qui doivent être examinées par chaque liaison de protocole:

- spécifier trois éléments d'informations d'identification: un URI qui identifie de façon univoque la liaison de protocole, des informations de contact postal ou électronique sur l'auteur, et une référence aux liaisons ou profils précédemment définis que la nouvelle liaison met à jour ou rend obsolètes;
- décrire l'ensemble des interactions entre parties impliquées dans la liaison. Toutes restrictions sur les applications utilisées par chaque partie et les protocoles impliqués dans chaque interaction doivent être explicitement cités;
- identifier les parties impliquées dans chaque interaction, y compris le nombre de parties impliquées et si des intermédiaires peuvent être impliqués;
- spécifier la méthode d'authentification des parties impliquées dans chaque interaction, y compris si l'authentification est exigée et les types d'authentification acceptables;
- identifier le niveau de soutien pour l'intégrité du message, y compris les mécanismes utilisés pour assurer l'intégrité du message;
- identifier le niveau de soutien pour la confidentialité, y compris si une tierce partie peut voir le contenu des messages et assertions SAML, si la liaison exige la confidentialité et les mécanismes recommandés pour réaliser la confidentialité;
- identifier les états d'erreur, y compris les états d'erreur chez chaque participant, spécialement ceux qui reçoivent et traitent les assertions ou messages SAML;
- identifier les considérations de sécurité, y compris l'analyse des menaces et la description des contre-mesures;
- identifier les considérations sur les métadonnées, telles que celles qui prennent en charge une liaison impliquant un protocole de communications particulier, ou utilisée dans un profil particulier, puissent être averties d'une façon efficace et interopérable.

10.2 Liaisons de protocole

Les paragraphes suivants définissent les liaisons de protocole qui sont spécifiées au titre de la norme SAML.

10.2.1 Considérations générales

Les paragraphes suivants décrivent les caractéristiques de toutes les liaisons de protocole définies pour SAML.

10.2.1.1 Utilisation de RelayState

Certaines liaisons définissent un mécanisme "RelayState" pour préserver et convoyer les informations d'état. Lorsqu'un tel mécanisme est utilisé pour convoyer un message de demande comme étape initiale d'un protocole SAML, il pose des exigences sur le choix et l'utilisation de la liaison utilisée ensuite pour convoyer la réponse. A savoir que si un message de demande SAML est accompagné de données RelayState, le répondant SAML doit alors retourner sa réponse de protocole SAML en utilisant une liaison qui accepte aussi un mécanisme RelayState, et il doit placer les données RelayState exactes qu'il a reçues avec la demande dans le paramètre RelayState correspondant dans la réponse.

10.2.1.2 Sécurité

Sauf mention contraire, ces déclarations de sécurité s'appliquent à toutes les liaisons. Les liaisons peuvent aussi faire des déclarations supplémentaires sur ces caractéristiques de sécurité.

1) Utilisation de TLS 1.0

Sauf mention contraire, dans toute utilisation de TLS 1.0 (RFC 2246 de l'IETF) par une liaison SAML, les serveurs doivent s'authentifier auprès des clients en utilisant un certificat X.509 v3. Le client doit établir l'identité du serveur sur la base du contenu du certificat (normalement par l'examen du champ DN de sujet du certificat, de l'attribut `subjectAltName`, etc.).

2) Authentification de l'origine des données

L'authentification du demandeur SAML et du répondant SAML associé au message est facultative et dépend de l'environnement d'utilisation. L'authentification des mécanismes disponibles à la couche d'échange de messages SOAP ou à partir du protocole de sous strate sous-jacente (par exemple, dans de

nombreuses liaisons, le protocole TLS ou HTTP) peut être utilisé pour fournir l'authentification de l'origine des données.

L'authentification du transport ne satisfera pas aux exigences d'authentification d'origine de bout en bout dans les liaisons où le message de protocole SAML passe par un intermédiaire – dans ce cas, l'authentification de message est recommandée.

SAML offre lui-même des mécanismes aux parties pour s'authentifier l'une l'autre, mais en plus, SAML peut utiliser d'autres mécanismes d'authentification pour fournir la sécurité pour SAML lui-même.

3) Intégrité de message

L'intégrité du message à la fois des demandes SAML et des réponses SAML est facultative et dépend de l'environnement d'utilisation. La couche sécurité dans le protocole de sous strate sous-jacente, ou un mécanisme à la couche d'échange de messages SOAP, peut être utilisé pour assurer l'intégrité du message.

L'intégrité du transport ne satisfera pas aux exigences d'authentification d'origine de bout en bout dans les liaisons où le message de protocole SAML passe par un intermédiaire – dans ce cas, l'authentification de message est recommandée.

4) Confidentialité du message

La confidentialité du message à la fois des demandes SAML et des réponses SAML est facultative et dépend de l'environnement d'utilisation. La couche sécurité dans le protocole de sous strate sous-jacente, ou un mécanisme à la couche d'échange de messages SOAP, peut être utilisé pour assurer la confidentialité du message.

La confidentialité du transport ne satisfera pas aux exigences de confidentialité de bout en bout dans les liaisons où le message de protocole SAML passe par un intermédiaire.

5) Autres considérations de sécurité

Avant le développement, chaque combinaison des mécanismes d'authentification, d'intégrité de message, et de confidentialité devrait être analysée quant à sa vulnérabilité dans le contexte spécifique de l'échange de protocole et de l'environnement de développement (voir les détails à l'Appendice I). La RFC 2617 de l'IETF décrit les attaques possibles dans l'environnement HTTP lorsque des schémas d'authentification de base ou de résumé de message sont utilisés. Un soin particulier devrait être apporté à l'impact possible de la mise en mémoire cache sur la sécurité.

10.2.2 Liaison SOAP SAML

SOAP est un protocole léger destiné à l'échange d'informations structurées dans un environnement décentralisé, et distribué. Il utilise les technologies XML pour définir un cadre d'échange de messages extensible fournissant une construction de messages qui peuvent être échangés sur divers protocoles sous-jacents. Le cadre a été conçu pour être indépendant de tout modèle de programmation particulier et autre sémantique spécifique de l'implémentation. Deux objectifs de conception majeurs pour SOAP sont la simplicité et l'extensibilité. SOAP essaye de satisfaire à ces objectifs en omettant, à partir du cadre d'échange de messages, les caractéristiques qu'on trouve souvent dans les systèmes distribués. De telles caractéristiques incluent, sans s'y limiter, la "fiabilité", la "sécurité", la "corrélation", "l'acheminement", et les "schémas d'échange de messages" (MEP, *message exchange pattern*).

Un message SOAP est fondamentalement une transmission unidirectionnelle entre des nœuds SOAP d'un expéditeur SOAP à un receveur SOAP, qui peut être acheminée à travers un ou plusieurs intermédiaires SOAP. On s'attend à ce que les messages SOAP soient combinés par application pour implémenter des schémas d'interactions plus complexes allant de la demande/réponse à des échanges "conversationnels" multiples, en va et vient.

SOAP définit une enveloppe de message XML qui inclut des sections d'en-tête et de corps, permettant aux données et aux informations de commande d'être transmises. SOAP définit aussi des règles de traitement associées à cette enveloppe et une liaison HTTP pour la transmission de messages SOAP.

La liaison SOAP SAML définit comment utiliser SOAP pour envoyer et recevoir les demandes et les réponses SAML.

Comme SAML, SOAP peut être utilisé sur plusieurs transports sous-jacents. Cette liaison a des aspects indépendants du protocole, mais appelle aussi l'utilisation de SOAP sur HTTP en tant que de besoin (implémentation obligatoire).

10.2.2.1 Informations requises

Identification: `urn:oasis:names:tc:SAML:2.0:bindings:SOAP`

Informations de contact: `security-services-comment@lists.oasis-open.org`

Description: donnée ci-dessous.

Mises à jour: `urn:oasis:names:tc:SAML:1.0:bindings:SOAP-binding`

10.2.2.2 Aspects indépendants du protocole de la liaison SOAP SAML

Les paragraphes qui suivent définissent les aspects de la liaison SOAP SAML qui sont indépendants du protocole sous-jacent, tels que HTTP, sur lequel sont transportés les messages SOAP. Cette liaison n'accepte que l'utilisation de SOAP 1.1.

10.2.2.2.1 Fonctionnement de base

Les messages SOAP 1.1 comportent trois éléments: une enveloppe, des données d'en-tête, et un corps de message. Les éléments de protocole de demande-réponse SAML doit être enclos dans le corps de message SOAP.

SOAP 1.1 définit aussi un système facultatif de codage des données. Ce système n'est pas utilisé dans la liaison SOAP SAML. Cela signifie que les messages SAML peuvent être transportés en utilisant SOAP sans recodage à partir du schéma SAML "standard" vers un schéma fondé sur le codage SOAP.

Le modèle de système utilisé pour les conversations SAML sur SOAP est un modèle simple de demande-réponse.

- Une entité système agissant comme demandeur SAML transmet un élément de demande SAML au sein du corps d'un message SOAP à une entité système agissant comme répondant SAML. Le demandeur SAML ne doit pas inclure plus d'une demande SAML par message SOAP ou inclure d'éléments XML additionnels dans le corps SOAP.
- Le répondant SAML doit retourner un élément de réponse SAML dans le corps d'un autre message SOAP ou générer une faute SOAP. Le répondant SAML ne doit pas inclure plus d'une réponse SAML par message SOAP ou inclure d'élément XML additionnel dans le corps SOAP. Si un répondant SAML ne peut pas, pour une raison quelconque, traiter une demande SAML, il doit générer une faute SOAP. Les codes de faute SOAP ne doivent pas être envoyés pour des erreurs au sein du domaine de problème SAML, par exemple, l'incapacité à trouver un schéma d'extension ou comme signal que le sujet n'est pas autorisé à accéder à une ressource dans une interrogation d'autorisation.

  NOTE (informative) – PE19 (voir OASIS PE:2006) suggère de remplacer l'alinéa ci-dessus par:

  Le répondant SAML devrait retourner un message SOAP contenant un élément de réponse SAML dans son corps ou une faute SOAP. Le répondant SAML ne doit pas inclure plus d'une réponse SAML par message SOAP ni inclure d'élément XML additionnel dans le corps SOAP. Les codes de faute SOAP ne devraient pas être envoyés pour des erreurs au sein du domaine de problème SAML, par exemple, l'incapacité à trouver un schéma d'extension ou comme signal que le sujet n'est pas autorisé à accéder à une ressource dans une interrogation d'autorisation.

A réception d'une réponse SAML dans un message SOAP, le demandeur SAML ne doit pas envoyer un code de faute ou d'autre message d'erreur au répondant SAML. Comme le format pour l'échange de message est un simple schéma demande-réponse, l'ajout d'éléments additionnels comme de conditions d'erreur compliquerait indûment le protocole.

SOAP du W3C fait référence à un projet de spécification du schéma XML qui inclut un espace de nom obsolète. Les demandeurs SAML devraient générer des documents SOAP ne faisant référence qu'à l'espace de nom de schéma final XML. Les répondants SAML doivent être capables de traiter les deux espaces de nom de schéma XML utilisés dans SOAP 1.1 (voir SOAP du W3C) ainsi que l'espace de nom de schéma final XML.

10.2.2.2.2 En-têtes SOAP

Un demandeur SAML dans une conversation SAML sur SOAP peut ajouter des en-têtes arbitraires au message SOAP. Cette liaison ne définit aucun en-tête SOAP supplémentaire.

NOTE 1 – La raison pour laquelle d'autres en-têtes ont besoin d'être admis est que certains logiciels et librairies SOAP pourraient ajouter des en-têtes à un message SOAP qui seraient hors du contrôle du processus SAML. Aussi, certains en-têtes pourraient être nécessaires pour les protocoles sous-jacents qui exigent l'acheminement des messages ou pour des mécanismes de sécurité des messages.

Un répondant SAML ne doit pas exiger d'en-tête dans le message SOAP pour le traitement correct du message SAML lui-même, mais peut exiger des en-têtes additionnels qui visent l'acheminement sous jacent ou les exigences de sécurité du message.

NOTE 2 – La raison en est qu'exiger des en-têtes supplémentaires causerait la fragmentation de la norme SAML et léserait l'interopérabilité.

10.2.2.3 Utilisation de SOAP sur HTTP

Un traitement SAML qui revendique la conformité à la liaison SOAP SAML doit implémenter SAML sur SOAP avec HTTP. Le présent paragraphe décrit certaines spécificités de l'utilisation de SOAP sur HTTP, y compris les en-têtes HTTP, la mise en mémoire cache, et le rapport d'erreurs.

La liaison HTTP pour SOAP est décrite au § 6.0 de SOAP du W3C. Elle requiert l'utilisation d'un en-tête `SOAPAction` au titre d'une demande HTTP SOAP. Un répondant SAML ne doit pas dépendre de la valeur de cet en-tête. Un demandeur SAML peut régler la valeur de l'en-tête SOAPAction comme suit:

```
http://www.oasis-open.org/committees/security
```

10.2.2.3.1 En-têtes HTTP

Un demandeur SAML dans une conversation SAML sur SOAP par HTTP peut ajouter des en-têtes arbitraires à la demande HTTP.Cette liaison ne définit aucun en-tête HTTP supplémentaire.

NOTE 1 – La raison pour laquelle d'autres en-têtes ont besoin d'être admis est que certains logiciels et librairies SOAP pourraient ajouter des en-têtes à un message SOAP qui seraient hors du contrôle du processus SAML. Aussi, certains en-têtes pourraient être nécessaires pour les protocoles sous jacents qui exigent l'acheminement des messages ou pour des mécanismes de sécurité des messages.

Un répondant SAML ne doit pas exiger d'en-tête dans le message SOAP pour le traitement correct du message SAML lui-même, mais peut exiger des en-têtes additionnels qui visent l'acheminement sous-jacent ou les exigences de sécurité du message.

NOTE 2 – La raison en est qu'exiger des en-têtes supplémentaires causerait la fragmentation de la norme SAML et léserait l'interopérabilité.

10.2.2.3.2 Mise en mémoire cache

Les mandataires HTTP ne devraient pas mettre en mémoire cache les messages de protocole SAML. Pour s'en assurer, les règles suivantes devrait être suivies.

Lors de l'utilisation de HTTP 1.1, les demandeurs devraient:

- inclure un champ d'en-tête `Cache-Control` réglé à "`no-cache, no-store`" *(pas de mémoire cache, pas de mémorisation)*;
- inclure un champ d'en-tête `Pragma` réglé à "`no-cache`" *(pas de mémoire cache)*.

Lors de l'utilisation de HTTP 1.1, les répondants devraient:

- inclure un champ d'en-tête `Cache-Control` réglé à "`no-cache, no-store, must-revalidate, private`" *(pas de mémoire cache, pas de mémorisation, doit revalider, privé)*;
- inclure un champ d'en-tête `Pragma` réglé à "`no-cache`";
- ne pas inclure de valideur, tel qu'un en-tête `Last-Modified` ou ETag.

10.2.2.3.3 Rapport d'erreurs

Un répondant SAML qui refuse d'effectuer un échange de messages avec le demandeur SAML devrait retourner une réponse "`403 Forbidden`". Dans ce cas, le contenu du corps HTTP n'est pas significatif

Comme décrit au paragraphe 6.2 de SOAP du W3C, dans le cas d'une erreur SOAP durant le traitement d'une demande SOAP, le serveur HTTP SOAP doit retourner une réponse "`500 Internal Server Error` (Erreur interne du serveur)" et inclure un message SOAP dans la réponse avec un élément `<SOAP-ENV:fault>` SOAP. Ce type d'erreur devrait être retourné pour les erreurs qui se rapportent à SOAP détectées avant que le contrôle ne soit passé au processeur SAML, ou quand le processeur SOAP rapporte une erreur interne (par exemple, l'espace de nom XML SOAP est incorrect, le schéma SAML ne peut pas être localisé, le processeur SAML soulève une exception, et ainsi de suite).

NOTE (informative) – PE19 (voir [OASIS Document Errata]) suggère de remplacer la première phrase de l'alinéa ci-dessus par:

> Comme décrit au paragraphe 6.2 de SOAP du W3C, dans le cas d'une erreur SOAP pendant le traitement d'une demande SOAP, le serveur HTTP SOAP devrait retourner une réponse "`500 Internal Server Error`" et inclure un message SOAP dans la réponse avec un élément `<SOAP-ENV:fault>` SOAP.

Dans le cas d'une erreur de traitement SAML, le serveur HTTP SOAP doit répondre par "`200 OK`" et inclure un élément `<samlp:Status>` spécifié par SAML dans la réponse SOAP au sein du corps SOAP. L'élément `<samlp:Status>` n'apparaît pas par lui-même dans le corps SOAP, mais seulement au sein d'une réponse SAML de quelque sorte.

Pour des informations complémentaires sur l'utilisation des codes d'état SAML, voir le paragraphe Assertions et protocoles SAML dans la présente Recommandation.

10.2.2.3.4 Considérations sur les métadonnées

La prise en charge de la liaison SOAP devrait être reflétée par l'indication d'un point d'extrémité d'URL auquel la demande contenue dans les messages SOAP pour un protocole ou profil particulier est à envoyer, ou autrement avec une définition d'accès/point d'extrémité WSDL.

10.2.2.3.5 Exemple d'échange de messages SAML utilisant SOAP sur HTTP

Ci-après figure un exemple d'interrogation qui demande une assertion contenant une déclaration d'attribut de la part d'une autorité d'attribut SAML.

```
POST /SamlService HTTP/1.1
Host: www.example.com
Content-Type: text/xml
Content-Length: nnn
SOAPAction: http://www.oasis-open.org/committees/security
<SOAP-ENV:Envelope
    xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
    <SOAP-ENV:Body>
        <samlp:AttributeQuery xmlns:samlp:="…"
xmlns:saml="…" xmlns:ds="…" ID="_6c3a4f8b9c2d" Version="2.0"
IssueInstant="2004-03-27T08:41:00Z"
            <ds:Signature> … </ds:Signature>
            <saml:Subject>
            …
            </saml:Subject>
        </samlp:AttributeQuery>
    </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
Following is an example of the corresponding response, which supplies an
assertion containing the attribute statement as requested.
HTTP/1.1 200 OK
Content-Type: text/xml
Content-Length: nnnn
<SOAP-ENV:Envelope
    xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
    <SOAP-ENV:Body>
        <samlp:Response xmlns:samlp="…" xmlns:saml="…" xmlns:ds="…"
ID="_6c3a4f8b9c2d" Version="2.0" IssueInstant="2004-03-27T08:42:00Z">
            <saml:Issuer>https://www.example.com/SAML</saml:Issuer>
            <ds:Signature> … </ds:Signature>
            <Status>
              <StatusCode Value="…"/>
            </Status>

            <saml:Assertion>
                <saml:Subject>
                …
                </saml:Subject>
                <saml:AttributeStatement>
            …
                </saml:AttributeStatement>
             </saml:Assertion>
        </samlp:Response>
    </SOAP-Env:Body>
</SOAP-ENV:Envelope>
```

10.2.3 Liaison SOAP inversée (PAOS)

Cette liaison renforce la spécification Liaison HTTP inversée pour SOAP (voir PAOS:2003). Les développeurs doivent se conformer aux règles générales de traitement spécifiées dans PAOS en plus de celles spécifiées dans la présente Recommandation. En cas de conflit, Liberty Allaiance POAS:2003 est la norme.

10.2.3.1 Informations requises

Identification: `urn:oasis:names:tc:SAML:2.0:bindings:PAOS`

Informations de contact: `security-services-comment@lists.oasis-open.org`

Description: donnée ci-dessous

Mises à jour: aucune.

10.2.3.2 Aperçu général

La liaison SOAP inversée est un mécanisme par lequel un demandeur HTTP peut faire connaître sa capacité à agir comme répondant SOAP ou comme intermédiaire SOAP à un demandeur SAML. Le demandeur HTTP est capable de

prendre en charge un schéma dans lequel une demande SAML lui est envoyée dans une enveloppe SOAP dans une réponse HTTP provenant du demandeur SAML, et le demandeur HTTP répond avec une réponse SAML dans une enveloppe SOAP dans une demande HTTP ultérieure. Ce schéma d'échange de messages accepte le cas d'utilisation défini dans le profil ECP SSO, dans lequel le demandeur HTTP est un intermédiaire dans un échange d'authentification.

10.2.3.3 Échange de messages

La liaison PAOS inclut deux schémas d'échange de messages de composant:

1) le demandeur HTTP envoie une demande HTTP à un demandeur SAML. Le demandeur SAML répond par une réponse HTTP contenant une enveloppe SOAP qui contient un message de demande SAML;
2) ensuite, le demandeur HTTP envoie une demande HTTP au demandeur SAML d'origine qui contient une enveloppe SOAP contenant un message de réponse SAML. Le demandeur SAML répond avec une réponse HTTP, éventuellement en réponse à la demande de service originale de l'étape 1.

Le profil ECP utilise la liaison PAOS pour fournir l'authentification du client au fournisseur de service avant que le service ne soit fourni. Cela intervient dans les étapes suivantes, illustrées à la Figure 10-1.

1) Le client demande un service en utilisant une demande HTTP.
2) Le fournisseur de service répond par une demande d'authentification SAML. Elle est envoyée en utilisant une demande SOAP, portée dans la réponse HTTP.
3) Le client retourne une réponse SOAP portant une réponse d'authentification SAML. Elle est envoyée en utilisant une nouvelle demande HTTP.
4) En supposant que l'authentification et l'autorisation du fournisseur de service sont réussies, le fournisseur de service peut répondre à la demande de service d'origine dans la réponse HTTP.

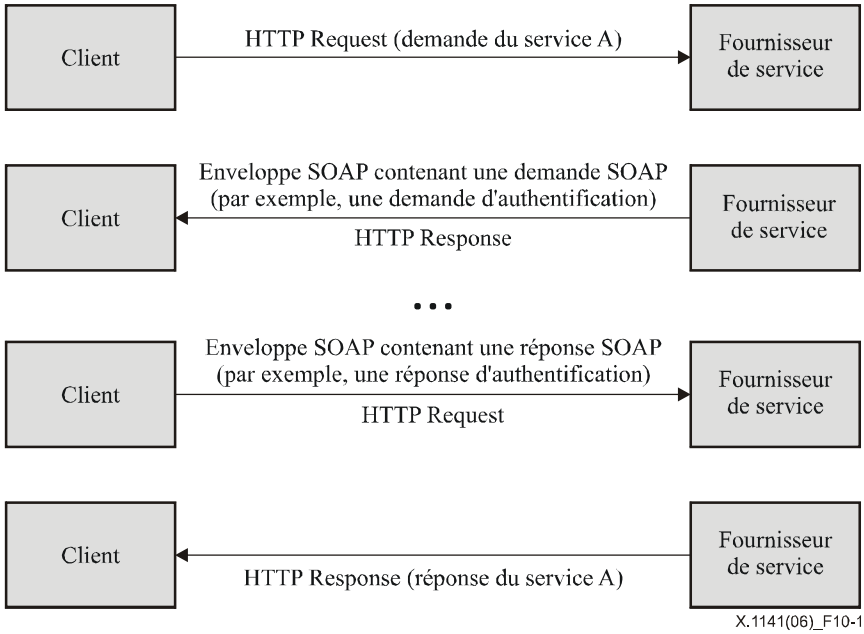

Figure 10-1/X.1141 – Échanges de messages de liaison PAOS

Le demandeur HTTP fait connaître sa capacité à traiter cette liaison SOAP inversée dans ses demandes HTTP en utilisant les en-têtes HTTP définis par la spécification PAOS:2003. En particulier:

- Le champ d'en-tête HTTP `Accept` doit indiquer la capacité à accepter le type de contenu "`application/vnd.paos+xml`".
- Le champ d'en-tête HTTP `PAOS` doit être présent et spécifier la version PAOS avec au minimum "`urn:liberty:paos:2003-08`".

NOTE 1 (informative) – PE21 (voir OASIS PE:2006) suggère de supprimer "au minimum" du texte ci-dessus.

Des en-têtes PAOS supplémentaires, tels que la valeur du service, peuvent être spécifiés par des profils qui utilisent la liaison PAOS. Le demandeur HTTP peut ajouter des en-têtes arbitraires à la demande HTTP.

NOTE 2 – Cette liaison ne définit pas de mécanisme RelayState. Les profils spécifiques qui utilisent cette liaison doivent donc définir un tel mécanisme, si besoin est. L'utilisation d'un en-tête SOAP est suggérée pour ce faire.

Les paragraphes suivants précisent les deux étapes de l'échange de messages.

10.2.3.3.1 Demande HTTP, demande SAML dans une réponse SOAP

En réponse à une demande HTTP arbitraire, le répondeur HTTP peut retourner un message de demande SAML en utilisant cette liaison en retournant une enveloppe SOAP 1.1 dans la réponse HTTP qui contient un seul message de demande SAML dans le corps SOAP, sans contenu de corps supplémentaire. L'enveloppe SOAP peut contenir des en-têtes SOAP arbitraires définis par PAOS, par des profils SAML, ou par des Recommandations supplémentaires.

Alors que le message de demande SAML est livré au demandeur HTTP, le receveur réel prévu peut être une autre entité système, le demandeur HTTP agissant comme intermédiaire, comme défini par des profils spécifiques.

10.2.3.3.2 Réponse SAML dans une demande SOAP, réponse HTTP

Lorsque le demandeur HTTP livre un message de réponse SAML au receveur prévu en utilisant la liaison PAOS, il le place comme seul élément du corps SOAP dans une enveloppe SOAP au sein d'une demande HTTP. Le demandeur HTTP peut être ou non à l'origine de la réponse SOAP. L'enveloppe SOAP peut contenir des en-têtes SOAP arbitraires définis par PAOS, par des profils SAML, ou par des Recommandations supplémentaires. L'échange SAML est considéré comme terminé et la réponse HTTP n'est pas spécifiée par cette liaison.

Des profils peuvent définir des contraintes supplémentaires sur le contenu HTTP des réponses non-SOAP durant les échanges couverts par cette liaison.

10.2.3.4 Mise en mémoire cache

Les mandataires HTTP ne devraient pas mettre en mémoire cache les messages de protocole SAML. Pour s'en assurer, les règles suivantes devrait être suivies.

En utilisant HTTP 1.1, les demandeurs qui envoient des messages de protocole SAML devraient:

- inclure un champ d'en-tête `Cache-Control` réglé à "`no-cache, no-store`";
- inclure un champ d'en-tête `Pragma` réglé à "`no-cache`".

En utilisant HTTP 1.1, les répondants qui retournent des messages de protocole SAML devraient:

- inclure un champ d'en-tête `Cache-Control` réglé à "`no-cache, no-store`, `must-revalidate`, `private`";
- inclure un champ d'en-tête `Pragma` réglé à "`no-cache`";
- Ne pas inclure de valideur, tel qu'un en-tête `Last-Modified` ou ETag.

10.2.3.5 Considérations sur la sécurité

Le demandeur HTTP dans la liaison PAOS peut agir comme un intermédiaire SOAP et lorsqu'il le fait, la sécurité de couche Transport pour l'authentification d'origine, l'intégrité et la confidentialité peuvent ne pas satisfaire aux exigences de sécurité de bout en bout. Dans ce cas, la sécurité à la couche de message SOAP est recommandée.

NOTE (informative) – PE31 (voir OASIS PE:2006) suggère de remplacer recommandée par RECOMMANDÉE.

10.2.3.5.1 Rapport d'erreurs

Les conventions d'erreur standards HTTP et SOAP doivent être observées. Les erreurs qui surviennent durant le taitement SAML ne doivent pas être signalées à la couche HTTP ou SOAP et doivent être traitées en utilisant des messages de réponse SAML avec un élément `<samlp:Status>` d'erreur.

10.2.3.5.2 Considérations sur les métadonnées

La prise en charge de la liaison PAOS devrait être reflétée par l'indication d'un point d'extrémité d'URL auquel les demandes HTTP et/ou les messages de protocole SAML contenus dans les enveloppes SOAP pour un protocole ou profil particulier sont à envoyer. Un seul point d'extrémité ou des points d'extrémité distincts pour la demande et la réponse peuvent être fournis.

10.2.4 Liaison HTTP redirect

La liaison HTTP Redirect définit un mécanisme par lequel les messages de protocole SAML peuvent être transmis au sein de paramètres d'URL. La longueur d'URL permise est théoriquement infinie, mais en pratique, limitée de façon imprévisible. Par conséquent, des codages spécialisés sont nécessaires pour porter les messages XML sur un URL, et des contenus de message plus longs ou plus complexes peuvent être envoyés en utilisant les liaisons HTTP POST ou Artifact.

Cette liaison peut être composée avec la liaison HTTP POST (voir au § 10.2.5) et la liaison HTTP Artifact (voir au § 10.2.6) pour transmettre les messages de demande et de réponse dans un seul échange de protocole utilisant deux liaisons différentes.

Cette liaison implique l'utilisation d'un codage de message. Bien que la définition de cette liaison inclut la définition d'un codage de message particulier, d'autres peuvent être définis et utilisés.

10.2.4.1 Informations requises

Identification: `urn:oasis:names:tc:SAML:2.0:bindings:HTTP-Redirect`

Informations de contact: `security-services-comment@lists.oasis-open.org`

Description: donnée ci-dessous.

Mises à jour: aucune.

10.2.4.2 Aperçu général

La liaison HTTP Redirect est destinée aux cas où le demandeur SAML et le répondant ont besoin de communiquer en utilisant comme intermédiaire un agent d'utilisateur HTTP (comme défini dans la RFC 2616 de l'IETF). Cela peut être nécessaire, par exemple, si les parties à la communication ne partagent pas un chemin de communication direct. Elle peut aussi être nécessaire si le répondant exige une interaction avec l'agent d'utilisateur afin de satisfaire la demande, comme lorsque l'agent d'utilisateur doit s'authentifier auprès de lui.

Certains agents d'utilisateur HTTP peuvent avoir la capacité de jouer un rôle plus actif dans l'échange de protocole et peuvent accepter d'autres liaisons qui utilisent HTTP, telles que les liaisons SOAP et SOAP inversé. Cette liaison ne suppose rien d'autre que les capacités d'un navigateur de la toile ordinaire.

10.2.4.3 RelayState

Les données RelayState peuvent être incluses dans un message de protocole SAML transmis avec cette liaison. Leur valeur ne doit pas excéder 80 octets en longueur et devrait être protégée en intégrité par l'entité qui crée le message indépendamment de toutes les autres protections qui peuvent ou non exister durant la transmission du message. La signature n'est pas réaliste étant donnée la limitation de place, mais, parce que la valeur est exposée à l'altération par un tiers, l'entité devrait s'assurer que la valeur n'a pas été altérée en utilisant une somme de contrôle, une valeur pseudo-aléatoire, ou des moyens similaires.

NOTE (informative) – PE1 (voir OASIS PE:2006) déclare que la dernière phrase de l'alinéa ci-dessus devrait se lire comme suit:

> Les données RelayState peuvent être incluses dans un message de protocole SAML transmis avec cette liaison. Leur valeur ne doit pas excéder 80 octets en longueur et devrait être protégée en intégrité par l'entité qui crée le message, via une signature numérique (voir § 10) ou par quelque moyen indépendant.

Si un message de demande SAML est accompagné de données RelayState, le répondant SAML doit alors retourner sa réponse de protocole SAML en utilisant une liaison qui accepte aussi le mécanisme RelayState, et il doit placer les données exactes qu'il a reçues avec la demande dans le paramètre RelayState correspondant dans la réponse.

Si une telle valeur n'est pas incluse avec un message de demande SAML, ou si le message de réponse SAML est généré sans une demande correspondante, le répondant SAML peut alors inclure des données RelayState pour qu'elles soient interprétées par le receveur sur la base de l'utilisation d'un profil ou d'un accord préalable entre les parties.

10.2.4.4 Codage du message

Les messages sont codés pour être utilisés avec cette liaison en utilisant une technique de codage d'URL, et transmis en utilisant la méthode HTTP GET. Il y a de nombreuses façons possibles de coder XML dans un URL, selon les contraintes en vigueur. La présente Recommandation définit une de ces méthodes, sans exclure les autres. Les points d'extrémité de liaison devraient indiquer quels codages ils acceptent en utilisant des métadonnées, en tant que de besoin. Les codages particuliers doivent être identifiés de façon univoque avec un URI lorsqu'il est défini. Il n'est pas exigé que tous les messages SAML possibles soient codables avec un ensemble de règles particulier, mais les règles doivent clairement indiquer quels messages ou contenus peuvent être ou non codés de cette manière.

Un codage d'URL doit placer entièrement le message au sein de la chaîne d'interrogation de l'URL, et doit réserver le reste de l'URL pour le point d'extrémité du receveur du message.

Un paramètre de chaîne d'interrogation nommé `SAMLEncoding` est réservé à l'identification du mécanisme de codage utilisé. Si ce paramètre est omis, la valeur est alors supposée être `urn:oasis:names:tc:SAML:2.0:bindings:URL-Encoding:DEFLATE`.

Tous les points d'extrémité qui prennent en charge cette liaison doivent accepter le codage DEFLATE décrit ci-après.

i) Codage DEFLATE

Identification: `urn:oasis:names:tc:SAML:2.0:bindings:URL-Encoding:DEFLATE`

Les messages de protocole SAML peuvent être codés dans un URL via la méthode de compression DEFLATE (RFC 1951 de l'IETF). Dans un tel codage, la procédure suivante devrait être appliquée à la sériation XML du message de protocole SAML original:

1) toute signature sur le message de protocole SAML, y compris l'élément `<ds:Signature>` XML lui-même, doit être retirée. Si le contenu du message inclut une autre signature, telle qu'une assertion SAML signée, cette signature enchassée n'est pas retirée. Cependant, la longueur d'un tel message après codage interdit d'utiliser ce mécanisme. Et donc, les messages de protocole SAML qui possèdent un contenu signé ne devraient pas être codés en utilisant ce mécanisme;
2) le mécanisme de compression DEFLATE, comme spécifié dans la RFC 1951 de l'IETF est alors appliqué à la totalité du contenu XML restant du message de protocole SAML original;
3) les données compressées sont ensuite codées en base64 conformément aux règles spécifiées dans la RFC 2045. Les renvois à la ligne (LF, *linefeed*) ou autres espaces blancs doivent être retirés du résultat;
4) les données codées en base64 sont alors codées en URL, et ajoutées à l'URL comme paramètre de chaîne d'interrogation qui doit être nommée `SAMLRequest` (si le message est une demande SAML) ou `SAMLResponse` (si le message est une réponse SAML);
5) si les données RelayState doivent accompagner le message de protocole SAML, elles doivent être codées en URL et placées dans un paramètre de chaîne d'interrogation supplémentaire nommée `RelayState`;
6) si le message de protocole SAML original était signé en utilisant une signature numérique XML, une nouvelle signature couvrant les données codées, comme spécifié ci-dessus, doit être jointe en utilisant les règles établies ci-dessous.

Les signatures numériques XML ne sont pas directement codées en URL conformément aux règles ci-dessus, du fait de problèmes d'espace. Si le message de protocole SAML sous-jacent est signé avec une signature XML, la forme codée en URL du message doit être signée comme suit:

1) l'identifiant d'algorithme de signature doit être inclus comme paramètre additionnel de chaîne d'interrogation, nommé `SigAlg`. La valeur de ce paramètre doit être un URI qui identifie l'algorithme utilisé pour signer le message de protocole SAML codé en URL, spécifié conformément à la signature XML ou à la Recommandation quelle qu'elle soit qui gouverne l'algorithme;
2) pour construire la signature, une chaîne consistant en la concaténation des paramètres de chaîne d'interrogation `RelayState` (s'il est présent), `SigAlg`, et `SAMLRequest` (ou `SAMLResponse`) (chacun étant codé en URL) est bâtie d'une des façons suivantes (ordonnée selon ce qui est indiqué ci-dessous):
   a) `SAMLRequest=value&RelayState=value&SigAlg=value`
      `SAMLResponse=value&RelayState=value&SigAlg=value`
   b) La chaîne d'octets résultante est la chaîne d'octets destinée à alimenter l'algorithme de signature. Aucun autre contenu de la chaîne d'interrogation d'origine n'est inclus ni signé.
   d) La valeur de signature doit être codée en utilisant le codage base64 (voir la RFC 2045 de l'IETF) avec retrait de tout espace blanc, et incluse comme un paramètre de chaîne d'interrogation nommé `Signature`. Certains caractères de la valeur de signature codés en base64 peuvent eux-mêmes requérir un codage en URL avant d'être ajoutés.
   c) Les algorithmes de signature suivants (voir Signature du W3C) et leurs représentations d'URI doivent être pris en charge avec ce mécanisme de codage:
      - DSAwithSHA1 – http://www.w3.org/2000/09/xmldsig#dsa-sha1
      - RSAwithSHA1 – http://www.w3.org/2000/09/xmldsig#rsa-sha1

   NOTE – L'Institut national US des normes et technologies (NIST, National Institute of Standards and Technology) encourage maintenant l'utilisation de SHA-256 (Algorithme de hachage sécurisé à clés codées de 256 bits) au lieu de SHA-1.

Lors de la vérification des signatures, l'ordre des paramètres de chaîne d'interrogation sur l'URL résultant à vérifier n'est pas prescrit par cette liaison. Les paramètres peuvent apparaître dans n'importe quel ordre. Avant de vérifier une signature, s'il en est, le consommateur d'assertions doit s'assurer que les valeurs de paramètres à vérifier sont ordonnées comme demandé par les règles de signature ci-dessus.

Le codage en URL n'est pas canonique; c'est-à-dire qu'il y a plusieurs codages légaux pour une valeur donnée. Le consommateur d'assertions doit donc effectuer les étapes de vérification en utilisant les valeurs originales codées en

URL qu'il a reçues sur la chaîne d'interrogation. Il n'est pas suffisant de recoder les paramètres après leur traitement par le logiciel parce que le codage résultant peut ne pas correspondre au codage du signataire.

S'il n'y a pas de valeur `RelayState`, tout le paramètre devrait être omis du calcul de signature (et non pas inclus comme nom de paramètre vide).

10.2.4.5 Échange de messages

Le modèle de système utilisé pour les conversations SAML via cette liaison est un modèle de demande-réponse, mais ces messages sont envoyés à l'agent d'utilisateur dans une réponse HTTP et délivrés au receveur du message dans une demande HTTP. Les interactions HTTP avant, pendant, et après que ces échanges aient lieu, sont non spécifiées. Le demandeur SAML et le répondant SAML sont tous deux supposés être des répondants HTTP. Voir le diagramme séquentiel suivant (Figure 10-2) qui illustre les messages échangés.

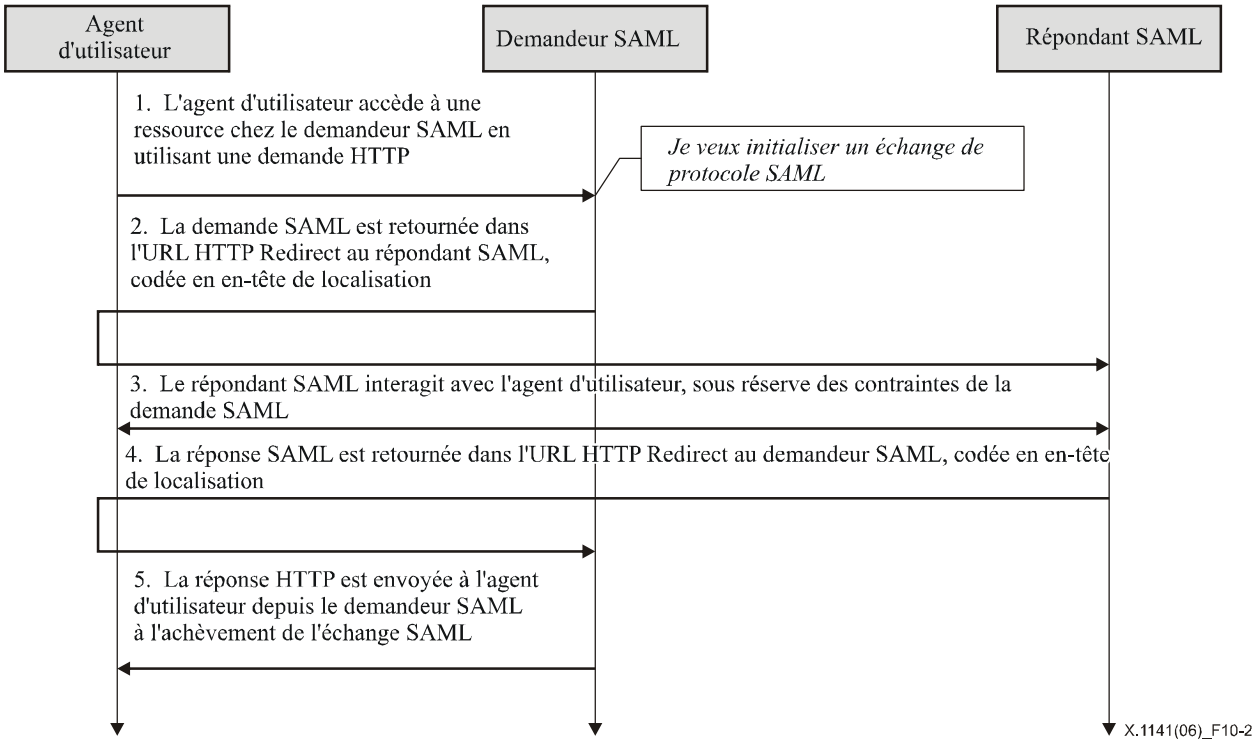

Figure 10-2/X.1141 – Échange de messages HTTP redirect

1) Au départ, l'agent d'utilisateur fait une demande HTTP arbitraire à une entité système. Dans le cours du traitement de la demande, l'entité système decide d'initialiser un échange de protocole SAML.
2) L'entité système agissant comme un demandeur SAML répond à la demande HTTP provenant de l'agent d'utilisateur à l'étape 1 en retournant une demande SAML. La demande SAML est retournée codée dans l'en-tête Localisation de la réponse HTTP, et l'état HTTP doit être 303 ou 302. Le demandeur SAML peut inclure une présentation et un contenu supplémentaires dans la réponse HTTP pour faciliter la transmission du message de l'agent d'utilisateur, comme défini dans la RFC 2616 de l'IETF. L'agent d'utilisateur délivre la demande SAML en produisant une demande HTTP GET au répondant SAML.
3) En général, le répondant SAML peut répondre à la demande SAML en retournant immédiatement une réponse SAML ou il peut retourner un contenu arbitraire pour faciliter les interactions ultérieures avec l'agent d'utilisateur qui seront nécessaires à la satisfaction de la demande. Des protocoles et profils spécifiques peuvent inclure des mécanismes pour indiquer le niveau d'incitation du demandeur à permettre cette sorte d'interaction (par exemple, l'attribut `IsPassive` dans `<samlp:AuthnRequest>`).
4) Finalement, le répondant devrait retourner une réponse SAML à l'agent d'utilisateur pour qu'elle soit retournée au demandeur SAML. La réponse SAML est retournée de la même façon que décrite pour la demande SAML à l'étape 2.
5) A réception de la réponse SOAP, le demandeur SAML retourne une réponse HTTP arbitraire à l'agent d'utilisateur.

10.2.4.5.1 HTTP et considérations sur les mémoires cache

Les mandataires HTTP et l'intermédiaire d'agent d'utilisateur ne devraient pas mettre en mémoire cache les messages de protocole SAML. Pour s'en assurer, les règles ci-après devraient être suivies.

En retournant les messages de protocole SAML à l'aide de HTTP 1.1, les répondants HTTP devraient:

- inclure un champ d'en-tête `Cache-Control` réglé à `"no-cache, no-store"`;
- inclure un champ d'en-tête `Pragma` réglé à "`no-cache`".

Il n'y a pas d'autre restriction à l'utilisation des en-têtes HTTP.

10.2.4.5.2 Considérations sur la sécurité

La présence de l'intermédiaire d'agent d'utilisateur signifie que le demandeur et le répondant ne peuvent s'appuyer sur la couche Transport pour l'authentification de bout en bout, l'intégrité et la confidentialité. Les messages codés en URL peuvent être signés pour fournir l'authentification d'origine et l'intégrité si la méthode de codage spécifie un moyen de signer.

Si le message est signé, l'attribut XML `Destination` dans l'élément racine SAML du message de protocole doit contenir l'URL auquel l'envoyeur a donné pour instruction à l'agent d'utilisateur de livrer le message. Le receveur doit alors vérifier que la valeur correspond à la localisation à laquelle le message a été reçu.

Cette liaison ne devrait pas être utilisée si le contenu de la demande ou réponse ne devrait pas être exposé à l'intermédiaire d'agent d'utilisateur. Autrement, la confidentialité des demandes et réponses SAML est facultative et dépend de l'environnement d'utilisation. Si la confidentialité est nécessaire, TLS 1.0 devrait être utilisé pour protéger le message en transit entre l'agent d'utilisateur et le demandeur et répondant SAML.

Les messages codés en URL peuvent être exposés dans divers enregistrements HTTP aussi bien que l'en-tête HTTP "Referrer".

Avant le développement, chaque combinaison des mécanismes d'authentification, d'intégrité de message, et de confidentialité devrait être analysée quant à sa vulnérabilité dans le contexte de l'échange de protocole spécifique, et de l'environnement de développement (voir l'Appendice I).

En général, cette liaison s'appuie sur l'authentification et la protection d'intégrité au niveau du message via la signature et ne prend pas en charge la confidentialité des messages provenant de l'intermédiaire d'agent d'utilisateur.

10.2.4.6 Rapport d'erreurs

Un répondant SAML qui refuse d'effectuer un échange de messages avec le demandeur SAML devrait retourner un message de réponse SAML avec une valeur `<samlp:StatusCode>` de second niveau de `urn:oasis:names:tc:SAML:2.0:status:RequestDenied`.

Les interactions HTTP durant l'échange de messages ne doivent pas utiliser les codes d'état d'erreur HTTP pour indiquer les échecs du traitement SAML, car l'agent d'utilisateur n'est pas un membre à part entière de l'échange de protocole SAML. Voir aussi au § 9.

10.2.4.7 Considérations sur les métadonnées

La prise en charge de la liaison HTTP Redirect devrait être reflétée par l'indication des points d'extrémité d'URL auxquels les demandes et réponses pour un protocole ou profil particulier devraient être envoyées. Un seul point d'extrémité ou des points d'extrémité distincts de demande et de réponse peuvent être fournis.

NOTE (informative) – PE2 (voir OASIS PE:2006) propose de remplacer l'alinéa ci-dessus par:

La prise en charge de la réception de messages utilisant la liaison HTTP Artifact devrait être reflétée par l'indication des points d'extrémité d'URL auxquels les demandes et réponses pour un protocole ou profil particulier devraient être envoyées. Un seul point d'extrémité ou des points d'extrémité de demande et de réponse distincts peuvent être fournis. La prise en charge de l'envoi des messages en utilisant cette liaison devrait être accompagnée par un ou plusieurs points d'extrémité `<md:ArtifactResolutionService>` indexés pour le traitement des messages `<samlp:ArtifactResolve>`.

10.2.4.8 Exemple d'échange de messages SAML utilisant HTTP Redirect

Dans cet exemple, une paire de messages `<LogoutRequest>` et `<LogoutResponse>` est échangée en utilisant la liaison HTTP Redirect.

D'abord, voici les messages de protocole SAML réels qui sont échangés:

```
<samlp:LogoutRequest xmlns:samlp="urn:oasis:names:tc:SAML:2.0:protocol"
xmlns="urn:oasis:names:tc:SAML:2.0:assertion"
    ID="d2b7c388cec36fa7c39c28fd298644a8" IssueInstant="2004-01-
21T19:00:49Z" Version="2.0">
    <Issuer>https://IdentityProvider.com/SAML</Issuer>
    <NameID Format="urn:oasis:names:tc:SAML:2.0:nameid-
format:persistent">005a06e0-ad82-110d-a556-004005b13a2b</NameID>
    <samlp:SessionIndex>1</samlp:SessionIndex>
</samlp:LogoutRequest>

<samlp:LogoutResponse xmlns:samlp="urn:oasis:names:tc:SAML:2.0:protocol"
xmlns="urn:oasis:names:tc:SAML:2.0:assertion"
    ID="b0730d21b628110d8b7e004005b13a2b"
InResponseTo="d2b7c388cec36fa7c39c28fd298644a8"
    IssueInstant="2004-01-21T19:00:49Z" Version="2.0">
    <Issuer>https://ServiceProvider.com/SAML</Issuer>
    <samlp:Status>
        <samlp:StatusCode
Value="urn:oasis:names:tc:SAML:2.0:status:Success"/>
    </samlp:Status>
</samlp:LogoutResponse>
```

La demande HTTP initiale provenant de l'agent d'utilisateur à l'étape 1 n'est pas définie par cette liaison. Pour initialiser l'échange de protocole de terminaison de session, le demandeur SAML retourne la réponse HTTP suivante, qui contient un message de demande SAML signé. La valeur de paramètre SAMLRequest est en fait déduite du message de demande ci-dessus. La portion signature n'est qu'une illustration et n'est pas le résultat d'un calcul réel. Les renvois à la ligne dans l'en-tête HTTP Location ci-dessous sont un artifice du document, et il n'y a pas de renvois à la ligne dans la valeur d'en-tête réelle.

```
HTTP/1.1 302 Object Moved
Date: 21 Jan 2004 07:00:49 GMT
Location:
https://ServiceProvider.com/SAML/SLO/Browser?SAMLRequest=fVFdS8MwFH0f7D%2BU
vGdNsq62oSsIQyhMESc%2B%2BJYlmRbWpObeyvz3puv2IMjyFM7HPedyK1DdsZdb%2F%2BEHfLF
fgwVMTt3RgTwzazIEJ72CFqRTnQWJWu7uH7dSLJjsg0ev%2FZFMlttiBWADtt6R%2BSyJr9msiR
H7O70sCm31Mj%2Bo%2BC%2B1KA5GlEWeZaogSQMw2MYBKodrIhjLKONU8FdeSsZkVr6T5M0GiHM
jvWCknqZXZ2OoPxF7kGnaGOuwxZ%2Fn4L9bY8NC%2By4du1XpRXnxPcXizSZ58KFTeHujEWkNPZ
ylsh9bAMYYUjO2Uiy3jCpTCMo5M1StVjmN9SO150sl9lU6RV2Dp0vsLIy7NM7YU82r9B90PrvCf
85W%2FwL8zSVQzAEAAA%3D%3D&RelayState=0043bfc1bc45110dae17004005b13a2b&SigAl
g=http%3A%2F%2Fwww.w3.org%2F200%2F09%2Fxmldsig%23rsa-
sha1&Signature=NOTAREALSIGNATUREBUTTHEREALONEWOULDGOHERE
Content-Type: text/html; charset=iso-8859-1
```

Après que des interactions non spécifiées ont pu avoir lieu, le répondant SAML retourne la réponse HTTP ci-dessous qui contient le message de réponse SAML signé. Encore une fois, la valeur du paramètre `SAMLResponse` est en fait déduite du message de réponse ci-dessus. La portion signature n'est qu'une illustration et non le résultat d'un calcul réel.

```
HTTP/1.1 302 Object Moved
Date: 21 Jan 2004 07:00:49 GMT
Location:
https://IdentityProvider.com/SAML/SLO/Response?SAMLResponse=fVFNa4QwEL0X%2B
h8k912TaDUGFUp7EbZQ6rKH3mKcbQVNJBOX%2FvxaXQ9tYec0vHlv3nzkqIZ%2BlAf7YSf%2FBj
hagxB8Db1BuZQKMjkjrcIOpVEDoPRa1o8vB8n3VI7OeqttT1bJbbJCBOc7a8j9XTBH9VyQhqYRb
TlrEi4Yo61oUqA0pvShYZHiDQkqs411tAVpeZPqSAgNOkrOas4zzcW55ZlI4liJrTXiBJVBr4wv
CJ877ijbcXZkmaRUxtk7CU7gcB5mLu8pKVddvghd%2Ben9iDIMa3CXTsOrs5euBbfXdgh%2F9sn
DK%2FEqW69Ye%2BUnvGL%2F8CfbQnBS%2FQS3z4QLW9aT1oBIws0j%2FGOyAb9%2FV34Dw5k779
IBAAA%3D&RelayState=0043bfc1bc45110dae17004005b13a2b&SigAlg=http%3A%2F%2Fww
w.w3.org%2F200%2F09%2Fxmldsig%23rsa-
sha1&Signature=NOTAREALSIGNATUREBUTTHEREALONEWOULDGOHERE
Content-Type: text/html; charset=iso-8859-1
```

10.2.5 Liaison HTTP POST

La liaison HTTP POST définit un mécanisme par lequel les messages de protocole SAML peuvent être transmis au sein du contenu codé en base64 d'un contrôle de forme HTML.

Cette liaison peut être composée avec la liaison HTTP Redirect (voir au § 10.2.4) et la liaison HTTP Artifact (voir au § 10.2.6) pour transmettre les messages de demande et de réponse dans un seul échange de protocole utilisant deux liaisons différentes.

10.2.5.1 Informations requises

Identification: `urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST`

Informations de contact: `security-services-comment@lists.oasis-open.org`

Description: donnée ci-dessous.

Mises à jour: aucune.

10.2.5.2 Généralités

La liaison HTTP POST est destinée aux cas où le demandeur et le répondant SAML ont besoin de communiquer en utilisant un agent d'utilisateur HTTP (comme défini dans la RFC 2616 de l'IETF) comme intermédiaire. Cela peut être nécessaire, par exemple, si les parties à la communication ne partagent pas un chemin direct de communication. Il peut aussi être nécessaire si le répondant exige une interaction avec l'agent d'utilisateur afin de satisfaire la demande, comme lorsque l'agent d'utilisateur doit s'authentifier auprès de lui.

Certains agents d'utilisateur HTTP peuvent avoir la capacité de jouer un rôle plus actif dans l'échange de protocole et peuvent accepter d'autres liaisons qui utilisent HTTP, telles que les liaisons SOAP et SOAP inversé. Cette liaison ne suppose rien d'autre que les capacités d'un navigateur de la toile ordinaire.

10.2.5.3 RelayState

Les données RelayState peuvent être incluses dans un message de protocole SAML transmis avec cette liaison. La valeur ne doit pas excéder 80 octets de long et devrait être protégée en intégrité par l'entité qui crée le message indépendamment de toute autre protection qui pourrait ou non exister durant la transmission du message. Signer n'est pas réaliste étant donné les limites en espace, mais comme la valeur est exposée à la manipulation d'un tiers, l'entité devrait s'assurer, en utilisant une somme de contrôle, une valeur pseudo aléatoire, ou des moyens similaires, que la valeur n'a pas été altérée.

Si un message de demande SAML est accompagné par des données RelayState, le répondant SAML doit alors retourner sa réponse de protocole SAML en utilisant une liaison qui accepte aussi un mécanisme RelayState, et il doit placer les données exactes qu'il a reçues avec la demande dans le paramètre RelayState correspondant dans la réponse.

Si aucune valeur de cette sorte n'est incluse avec un message de demande SAML, ou si le message de réponse SAML est généré sans une demande correspondante, le répondant SAML peut alors inclure des données RelayState à interpréter par le receveur sur la base de l'utilisation d'un profil ou d'un accord préalable entre les parties.

NOTE (informative) – PE31 (voir OASIS PE:2006) suggère de préciser l'alinéa précédent de la façon suivante:

Si aucun paramètre RelayState n'est inclus avec un message de demande SAML, ou si le message de réponse SAML est généré sans demande correspondante, le répondant SAML peut alors inclure des données RelayState à interpréter par le receveur sur la base de l'utilisation d'un profil ou d'un accord préalable entre les parties.

10.2.5.4 Codage de message

Les messages sont codés pour être utilisés avec cette liaison en codant le XML en un contrôle de forme HTML et sont transmis en utilisant la méthode HTTP POST. Un message de protocole SAML est codé en forme en appliquant les règles de codage de base64 à la représentation XML du message et en plaçant le résultat dans un contrôle à forme cachée au sein d'une forme, comme défini par le § 17 de HTML du W3C. Le document HTML doit adhérer au XHTML du W3C, conformément aux pratiques communes.

Si le message est une demande SAML, le contrôle de forme doit être dénommé `SAMLRequest`. Si le message est une réponse SAML, le contrôle de forme doit alors être nommé `SAMLResponse`. Tout contrôle de forme ou présentation supplémentaire peut être inclus mais ne doit pas être exigé pour que le receveur traite le message.

Si une valeur "RelayState" doit accompagner le message de protocole SAML, elle doit être placée dans un contrôle de forme caché supplémentaire nommé `RelayState` au sein de la même forme avec le message SAML.

L'attribut d'action de la forme doit être le point d'extrémité HTTP du receveur pour le protocole ou profil utilisant cette liaison à laquelle le message SAML doit être livré. L'attribut de méthode doit être "POST".

Toute technique prise en charge par l'agent d'utilisateur peut être utilisée pour causer la soumission de la forme, et tout contenu de forme nécessaire à sa prise en charge peut être inclus, comme des commandes de soumission et d'écriture du côté client. Cependant, le receveur doit être capable de traiter le message sans avoir à considérer les mécanismes par lesquels est initialisée la soumission de forme.

Toutes valeurs de contrôle de forme incluses doivent être transformées de façon à être bonnes à inclure dans le document XHTML. Cela inclut des caractères de transformation tels que des guillemets en entités HTML, etc.

10.2.5.5 Échange de messages

Le modèle de système utilisé pour les conversations SAML via cette liaison est un modèle de demande-réponse, mais ces messages sont envoyés à l'agent d'utilisateur dans une réponse HTTP et livrés au receveur du message dans une demande HTTP. Les interactions HTTP avant, pendant et après que ces échanges ont lieu sont non spécifiées. Le demandeur et le répondant SAML sont tous deux supposés être les répondants HTTP. Voir la Figure 10-3 qui illustre les messages échangés.

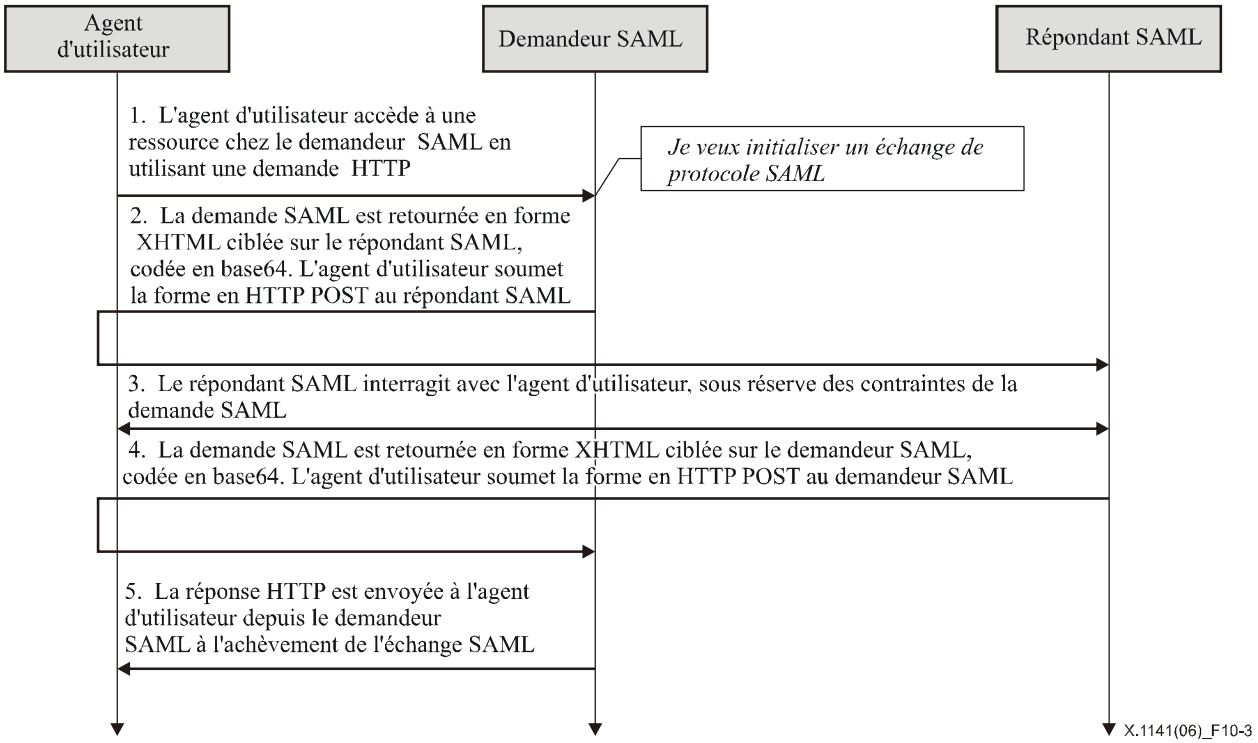

Figure 10-3/X.1141 – Échange de messages HTTP POST

1) D'abord, l'agent d'utilisateur fait une demande HTTP arbitraire à une entité système. Dans le cours du traitement de la demande, l'entité système décide d'initialiser un échange de protocole SAML.
2) L'entité système agissant comme demandeur SAML répond à une demande HTTP de l'agent d'utilisateur en retournant une demande SAML. La demande est retournée dans un document XHTML qui contient la forme et le contenu définis au § 10.2.5.4. L'agent d'utilisateur livre la demande SAML en produisant une demande HTTP POST au répondant SAML.
3) En général, le répondant SAML peut répondre à la demande SAML en retournant immédiatement une réponse SAML ou il peut retourner un contenu arbitraire pour faciliter l'interaction ultérieure avec l'agent d'utilisateur qui sera nécessaire pour satisfaire la demande. Des protocoles et profils spécifiques peuvent inclure des mécanismes pour indiquer le niveau d'acceptation du demandeur à permettre ce type d'interaction (par exemple, l'attribut `IsPassive` dans `<samlp:AuthnRequest>`).
4) Finalement le répondant devrait retourner une réponse SAML à l'agent d'utilisateur, à retourner au demandeur SAML. La réponse SAML est retournée de la même façon que celle décrite pour la demande SAML à l'étape 2.
5) A réception de la réponse SAML, le demandeur SAML retourne une réponse HTTP arbitraire à l'agent d'utilisateur.

10.2.5.5.1 HTTP et considérations de mise en mémoire cache

Les mandataires HTTP et l'intermédiaire d'agent d'utilisateur ne devraient pas mettre en mémoire cache les messages de protocole SAML. Pour s'en assurer, les règles suivantes devraient être suivies.

En retournant les messages de protocole SAML utilisant HTTP 1.1, les répondants HTTP devraient:

- inclure un champ d'en-tête `Cache-Control` réglé à "`no-cache`, `no-store`";
- inclure un champ d'en-tête `Pragma` réglé à "`no-cache`".

Il n'y a pas d'autre restriction à l'utilisation des en-têtes HTTP.

10.2.5.5.2 Considérations sur la sécurité

La présence de l'intermédiaire d'agent d'utilisateur signifie que le demandeur et le répondant ne peuvent pas s'appuyer sur la couche Transport pour l'authentification de bout en bout, la protection de l'intégrité ou de la confidentialité et doivent au lieu de cela authentifier les messages reçus. SAML fournit la signature sur les messages de protocole pour l'authentification et l'intégrité dans de tels cas. Les messages codés en forme peuvent être signés avant l'aplication du codage en base64.

Si le message est signé, l'attribut `Destination` XML dans l'élément racine SAML du message de protocole doit contenir l'URL auquel l'envoyeur a donné ordre à l'agent d'utilisateur de délivrer le message. Le receveur doit alors vérifier que la valeur correspond à la localisation à laquelle le message a été reçu.

Cette liaison ne devrait pas être utilisée si le contenu de la demande ou de la réponse ne devrait pas être exposé à l'intermédiaire de l'agent d'utilisateur. Autrement, la confidentialité des demandes et des réponses SAML est facultative et dépend de l'environnement d'utilisation. Si la confidentialité est nécessaire, TLS 1.0 devrait être utilisé pour protéger le message en transit entre l'agent d'utilisateur, le demandeur et le répondant SAML.

En général, cette liaison s'appuie sur l'authentification et la protection d'intégrité au niveau du message via la signature et ne prend pas en charge la confidentialité des messages provenant de l'intermédiaire d'agent d'utilisateur.

Aucun mécanisme n'est défini pour protéger l'intégrité des relations entre le message de protocole SAML et la valeur "RelayState", s'il en est. C'est-à-dire qu'un attaquant à la possibilité de recombiner une paire de réponses HTTP valides en commutant les valeurs "RelayState" associées à chaque message de protocole SAML. Les valeurs individuelles de "RelayState" et de message SAML peuvent être protégées en intégrité, pas la combinaison. Il en résulte que le producteur et le consommateur d'informations "RelayState" doivent veiller à ne pas associer d'informations d'état sensibles à la valeur "RelayState" sans prendre de précautions supplémentaires (telles que celles fondées sur les informations du message SAML).

10.2.5.6 Rapport d'erreurs

Un répondant SAML qui refuse d'effectuer un échange de messages avec le demandeur SAML devrait retourner un message de réponse avec une valeur `<samlp:StatusCode>` de second niveau de `urn:oasis:names:tc:SAML:2.0:status:RequestDenied`.

Les interactions HTTP durant l'échange de messages ne doivent pas utiliser des codes d'état d'erreur pour indiquer des échecs du traitement SAML, car l'agent d'utilisateur n'est pas membre à part entière de l'échange de protocole SAML.

Pour des informations complémentaires sur les codes d'état SAML, voir au § 8.2.

10.2.5.7 Considérations sur les métadonnées

La prise en charge de la liaison HTTP POST devrait être reflétée par l'indication des points d'extrémité d'URL auxquels les demandes et réponses pour un protocole ou profil particulier devraient être envoyées. Un seul point d'extrémité ou des points d'extrémité distincts pour la demande et la réponse peuvent être fournis.

10.2.5.8 Exemple d'échange de messages SAML utilisant HTTP POST

Dans cet exemple, une paire de messages `<LogoutRequest>` et `<LogoutResponse>` est échangée en utilisant la liaison HTTP POST.

Ici figurent d'abord les messages de protocole SAML réels qui sont échangés:

```
<samlp:LogoutRequest xmlns:samlp="urn:oasis:names:tc:SAML:2.0:protocol"
xmlns="urn:oasis:names:tc:SAML:2.0:assertion"
    ID="d2b7c388cec36fa7c39c28fd298644a8" IssueInstant="2004-01-
21T19:00:49Z" Version="2.0">
    <Issuer>https://IdentityProvider.com/SAML</Issuer>
    <NameID Format="urn:oasis:names:tc:SAML:2.0:nameid-
format:persistent">005a06e0-ad82-110d-a556-004005b13a2b</NameID>
    <samlp:SessionIndex>1</samlp:SessionIndex>
</samlp:LogoutRequest>
```

```
<samlp:LogoutResponse xmlns:samlp="urn:oasis:names:tc:SAML:2.0:protocol"
xmlns="urn:oasis:names:tc:SAML:2.0:assertion"
    ID="b0730d21b628110d8b7e004005b13a2b"
InResponseTo="d2b7c388cec36fa7c39c28fd298644a8"
    IssueInstant="2004-01-21T19:00:49Z" Version="2.0">
    <Issuer>https://ServiceProvider.com/SAML</Issuer>
    <samlp:Status>
        <samlp:StatusCode
Value="urn:oasis:names:tc:SAML:2.0:status:Success"/>
    </samlp:Status>
</samlp:LogoutResponse>
```

La demande HTTP initiale de l'agent d'utilisateur à l'étape 1 n'est pas définie par cette liaison. Pour initialiser l'échange de protocole de terminaison de session, le demandeur SAML retourne la réponse HTTP suivante, qui contient un message de demande SAML. La valeur de paramètre SAMLRequest est en fait déduite du message de demande ci-dessus.

```
HTTP/1.1 200 OK
Date: 21 Jan 2004 07:00:49 GMT
Content-Type: text/html; charset=iso-8859-1

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN"
"http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en">
<body onload="document.forms[0].submit()">

<noscript>
<p>
<strong>Note:</strong> Since your browser does not support JavaScript, you
must press the Continue button once to proceed.
</p>
</noscript>

<form action="https://ServiceProvider.com/SAML/SLO/Browser" method="post">
<div>
<input type="hidden" name="RelayState"
value="0043bfc1bc45110dae17004005b13a2b"/>
<input type="hidden" name="SAMLRequest"
value="PHNhbWxwOkxvZ291dFJlcXVlc3QgeG1sbnM6c2FtbHA9InVybjpvYXNpczpuYW1l
czp0YzpTQU1MOjIuMDpwcm90b2NvbCIgeG1sbnM9InVybjpvYXNpczpuYW1lczp0
YzpTQU1MOjIuMDphc3NlcnRpb24iDQogICAgSUQ9ImQyYjdjMzg4Y2VjMzZmYTdj
MzljMjhmZDI5ODY0NGE4IiBJc3N1ZUluc3RhbnQ9IjIwMDQtMDEtMjFUMTk6MDA6
NDlaIiBWZXJzaW9uPSIyLjAiPg0KICAgIDxJc3N1ZXI+aHR0cHM6Ly9JZGVudGl0
eVByb3ZpZGVyLmNvbS9TQU1MPC9Jc3N1ZXI+DQogICAgPE5hbWVJRCBBGb3JtYXQ9
InVybjpvYXNpczpuYW1lczp0YzpTQU1MOjIuMDpuYW1laWQtZm9ybWF0OnBlcnNp
c3RlbnQiPjAwNWEwNmUwLWFkODItMTEwZC1hNTU2LTAwNDAwNWIxM2EyYjwvTmFt
ZUlEPg0KICAgIDxzYW1scDpTZXNzaW9uSW5kZXg+MTwvc2FtbHA6U2Vzc2lvbklu
ZGV4Pg0KPC9zYW1scDpMb2dvdXRSZXF1ZXN0Pg=="/>
</div>
<noscript>
<div>
<input type="submit" value="Continue"/>
</div>
</noscript>
</form>
</body>
</html>
```

Après que toutes les interactions non spécifiées ont pu avoir lieu, le répondant SAML retourne la réponse HTTP ci-dessous qui contient le message de réponse SAML. A nouveau, la valeur de paramètre SAMLResponse est en fait déduite du message de réponse ci-dessus.

```
HTTP/1.1 200 OK
Date: 21 Jan 2004 07:00:49 GMT
Content-Type: text/html; charset=iso-8859-1

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN"
"http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">
```

```
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en">
<body onload="document.forms[0].submit()">

<noscript>
<p>
<strong>Note:</strong> Since your browser does not support JavaScript, you
must press the Continue button once to proceed.
</p>
</noscript>

<form action="https://IdentityProvider.com/SAML/SLO/Response"
method="post">
<div>
<input type="hidden" name="RelayState"
value="0043bfc1bc45110dae17004005b13a2b"/>
<input type="hidden" name="SAMLResponse"
value="PHNhbWxwOkxvZ291dFJlc3BvbnNlIHhtbG5zOnNhbWxwPSJ1cm46b2FzaXM6bmFt
ZXM6dGM6U0FNTDoyLjA6cHJvdG9jb2wiIHhtbG5zPSJ1cm46b2FzaXM6bmFtZXM6
dGM6U0FNTDoyLjA6YXNzZXJ0aW9uIg0KICAgIElEPSJiMDczMGQyMWI2MjgxMTBk
OGI3ZTAwNDAwNWIxM2EyYiIgSW5SZXNwb25zZVRvPSJkMmI3YzM4OGNlYzM2ZmE3
YzM5YzI4ZmQyOTg2NDRhOCINCiAgICBJc3N1ZUluc3RhbnQ9IjIwMDQtMDEtMjFU
MTk6MDA6NDlaIiBWZXJzaW9uPSIyLjAiPg0KICAgIDxJc3N1ZXI+aHR0cHM6Ly9T
ZXJ2aWNlUHJvdmlkZXIuY29tL1NBTUw8L0lzc3Vlcj4NCiAgICA8c2FtbHA6U3Rh
dHVzPg0KICAgICAgICA8c2FtbHA6U3RhdHVzQ29kZSBWYWx1ZT0idXJuOm9hc2lz
Om5hbWVzOnRjOlNBTUw6Mi4wOnN0YXR1czpTdWNjZXNzIi8+DQogICAgPC9zYW1s
cDpTdGF0dXM+DQo8L3NhbWxwOkxvZ291dFJlc3BvbnNlPg=="/>
</div>
<noscript>
<div>
<input type="submit" value="Continue"/>
</div>
</noscript>
</form>
</body>
</html>
```

10.2.6 Liaison HTTP Artifact

Dans la liaison HTTP Artifact, la demande SAML, la réponse SAML, ou les deux, sont transmises par référence en utilisant un petit substitut appelé un artifice. Une liaison séparée, synchrone, comme la liaison SOAP de SAML, est utilisée pour échanger l'artifice contre le message de protocole réel en utilisant le protocole de résolution d'artifice défini au § 8.

Cette liaison peut être composée avec la liaison HTTP Redirect (voir au § 10.2.4) et la liaison HTTP POST (voir au § 10.2.5) pour transmettre les messages de demande et de réponse dans un seul échange de protocole utilisant deux liaisons différentes.

10.2.6.1 Informations requises

Identification: `urn:oasis:names:tc:SAML:2.0:bindings:HTTP-Artifact`

Informations de contact: security-services-comment@lists.oasis-open.org

Description: donnée ci-dessous.

Mises à jour: aucune.

10.2.6.2 Aperçu général

La liaison HTTP Artifact est destinée aux cas dans lesquels le demandeur et le répondant SAML ont besoin de communiquer en utilisant un agent d'utilisateur HTTP comme intermédiaire, mais les limites de l'intermédiaire empêchent ou découragent la transmission d'un message (ou échange de messages) entier à travers lui. Cela peut être pour des raisons techniques ou parce qu'il a une réticence à exposer le contenu du message à l'intermédiaire (et si l'utilisation du chiffrement n'est pas pratique).

A cause de la nécessité de résoudre ultérieurement l'artifice en utilisant une autre liaison synchrone, comme SOAP, un chemin de communication direct doit exister entre l'envoyeur du message SAML et le receveur dans la direction inverse de la transmission de l'artifice (le receveur du message et de l'artifice doit être capable de renvoyer une demande `<samlp:ArtifactResolve>` au producteur de l'artifice). Le producteur de l'artifice doit aussi maintenir l'état pendant que l'artifice est en cours, ce qui a des implications pour les environnements à équilibrage de charge.

10.2.6.3 Codage du message

Il y a deux méthodes de codage d'un artifice à utiliser avec cette liaison. Une est de coder l'artifice dans un paramètre d'URL et l'autre est de placer l'artifice dans un contrôle de forme HTML. Lorsque le codage d'URL est utilisé, la méthode HTTP GET est utilisée pour délivrer le message, alors que POST est utilisé avec le codage de forme. Tous les points d'extrémité qui prennent en charge cette liaison doivent accepter les deux techniques.

10.2.6.3.1 RelayState

Les données RelayState peuvent être incluses avec un artifice SAML transmis avec cette liaison. La valeur ne doit pas excéder 80 octets de long et devrait être protégée en intégrité par l'entité qui crée le message indépendamment de toutes autres protections pouvant exister ou non durant la transmission du message. La signature n'est pas réaliste étant donné les limites d'espace, mais comme la valeur est exposée à des manipulations de tiers, l'entité devrait s'assurer que la valeur n'a pas été altérée en utilisant une somme de contrôle, une valeur pseudo-aléatoire, ou des moyens similaires.

Si un artifice représentant une demande SAML est accompagné de données RelayState, le répondant SAML doit alors retourner sa réponse de protocole SAML en utilisant une liaison qui prend aussi en charge un mécanisme RelayState, et il doit placer les données exactes qu'il a reçues avec l'artifice dans le paramètre RelayState correspondant dans la réponse.

Si aucune valeur de cette sorte n'est incluse avec un artifice représentant une demande SAML, ou si le message de réponse SAML est généré sans demande correspondante, le répondant SAML peut alors inclure des données RelayState à interpréter par le receveur sur la base de l'utilisation d'un profil ou d'un accord préalable entre les parties.

10.2.6.3.2 Codage d'URL

Pour coder un artifice dans un URL, la valeur d'artifice est codée en URL et placée dans un paramètre de chaîne d'interrogation nommée `SAMLart`.

Si une valeur "RelayState" doit accompagner l'artifice SAML, elle doit être codée en URL et placée dans un paramètre de chaîne d'interrogation supplémentaire nommé `RelayState`.

10.2.6.3.3 Codage de forme

Un artifice SAML est codé en forme en le plaçant dans un contrôle de forme caché au sein d'une forme, comme défini par HTML du W3C. Le document HTML doit adhérer à XHTML du W3C. Le contrôle de forme doit être nommé `SAMLart`. Tout contrôle de forme supplémentaire ou présentation peut être inclus mais ne doit pas être exigé pour que le receveur traite l'artifice.

Si une valeur "RelayState" doit accompagner l'artifice SAML, il doit être placé dans un contrôle de forme supplémentaire nommé `RelayState`, dans la même forme, avec le message SAML.

L'attribut d'action de la forme doit être le point d'extrémité HTTP du receveur pour le protocole ou le profil qui utilise cette liaison à laquelle l'artifice est à livrer. L'attribut de `méthode` doit être réglé à "POST".

Toute technique prise en charge par l'agent d'utilisateur peut être utilisée pour causer la soumission de la forme, et tout contenu de forme nécessaire à cette prise en charge peut être inclus, comme des contrôles de soumission et des commandes d'écriture côté client. Cependant, le receveur doit être capable de traiter l'artifice sans considération du mécanisme par lequel la soumission de forme est initialisée.

Toute valeur de contrôle de forme incluse doit être transformée de façon à être incluse en toute sécurité dans le document XHTML. Cela inclut de transformer des caractères tels que les guillemets en entités HTML, etc.

10.2.6.4 Format d'artifice

Par rapport à cette liaison, un artifice est une chaîne courte et opaque. Différents types peuvent être définis et utilisés sans affecter la liaison. Les caractéristiques importantes sont la capacité d'un receveur d'artifice à identifier le producteur de l'artifice, la résistance à l'altération et à la contrefaçon, l'unicité, et la compacité.

Le format général de tout artifice inclut un code de type d'artifice de deux octets obligatoire et une valeur d'indice de deux octets qui identifie un point d'extrémité spécifique du service de résolution de l'artifice du producteur, comme suit:

```
SAML_artifact      := B64(TypeCode EndpointIndex RemainingArtifact)
TypeCode           := Byte1Byte2
EndpointIndex      := Byte1Byte2
```

La notation `B64 (TypeCode EndpointIndex RemainingArtifact)` figure l'application de la transformation en base64 (voir la RFC 2045 de l'IETF) en concaténation de `TypeCode, EndpointIndex`, et `RemainingArtifact`.

Les pratiques suivantes sont recommandées pour la création des artifices SAML:

- un URI identifiant est attribué à chaque producteur, appelé aussi identifiant d'entité de producteur (ou de fournisseur). Voir au § 8 la discussion de ce type d'identifiant;
- le producteur construit le composant `SourceID` de l'artifice en prenant le hachage SHA-1 de l'URL d'identification. La valeur du hachage n'est pas codée en hexadécimal;

  NOTE 1 – L'Institut National US des normes et technologies (NIST) encourage maintenant l'utilisation de SHA-256 (Algorithme de hachage sécurisé à clés codées de 256 bits) à la place de SHA-1;
- la valeur `MessageHandle` est construite à partir d'une séquence de chiffres cryptographiquement fortement aléatoire ou pseudo-aléatoire (voir la RFC 1750 de l'IETF) générée par le producteur. La séquence consiste en valeurs d'au moins 16 octets de long. Ces valeurs devraient être bourrées en tant que de besoin pour une longueur totale de 20 octets.

  NOTE 2 (informative) – PE4 (voir [OASIS Errata Document]) suggère d'ajouter le texte suivant à la fin du paragraphe ci-dessus:

  Bien que la structure générale d'artifice ressemble à celle utilisée dans les versions précédentes de SAML et que le code de type du format unique décrit ci-dessous ne soit pas en conflit avec les formats précédemment définis, il n'y a explicitement aucune correspondance entre les artifices de SAML 2.0 et ceux qu'on trouve dans les spécifications précédentes, et les formats d'artifices non définis spécifiquement pour être utilisés avec SAML 2.0 ne doivent pas être utilisés avec cette liaison.

Ce qui suit décrit le type unique d'artifice défini par SAML V2.0.

10.2.6.4.1 Informations requises

Identification: `urn:oasis:names:tc:SAML:2.0:artifact-04`

Informations de contact: security-services-comment@lists.oasis-open.org

Description: donnée ci-dessous

Mises à jour: aucune

10.2.6.4.2 Détails du format

SAML V2.0 définit un type d'artifice codé 0x0004. Ce type d'artifice est défini comme suit:

```
TypeCode          := 0x0004
RemainingArtifact := SourceID MessageHandle
SourceID          := 20-byte_sequence
MessageHandle     := 20-byte_sequence
```

`SourceID` est une séquence de 20 octets utilisée par le receveur de l'artifice pour déterminer l'identité du producteur de l'artifice et l'ensemble des points d'extrémité de résolution possibles.

On suppose que le site de destination établira un tableau des valeurs de `SourceID` ainsi qu'un ou plusieurs points d'extrémité d'URL indexés (ou d'adresses) pour le répondant SAML correspondant. Le paragraphe 9 peut être utilisé à cette fin. A réception de l'artifice SAML, le receveur détermine si le `SourceID` appartient à un producteur d'artifice connu et obtient la localisation du répondant SAML en utilisant le `EndpointIndex` avant de lui envoyer un message SAML `<samlp:ArtifactResolve>`.

Deux producteurs d'artifice pour un receveur commun doivent utiliser des valeurs de `SourceID` distinctes. La construction des valeurs de `MessageHandle` est gouvernée par le principe qu'elles ne devraient avoir aucune relation prévisible avec le contenu du message référencé sur le site de production et il doit être irréalisable de construire ou deviner la valeur d'un message valide en cours.

10.2.6.5 Échange de messages

Le modèle de système utilisé pour les conversations SAML au moyen de cette liaison est un modèle de demande-réponse dans lequel une référence d'artifice prend la place du contenu réel du message, et la référence d'artifice est envoyée à l'agent d'utilisateur dans une réponse HTTP et délivrée au receveur du message dans une demande HTTP. Les interactions HTTP avant, pendant et après que ces échanges ont lieu sont non spécifiées. Le demandeur et le répondant SAML sont tous deux supposés être les répondants HTTP.

De plus, on suppose que à réception d'un artifice au moyen de l'agent d'utilisateur, le receveur invoque un échange séparé direct avec le producteur de l'artifice en produisant le protocole de résolution d'artifice défini dans la présente Recommandation. Cet échange doit utiliser une liaison sans intermédiaire d'agent d'utilisateur HTTP, telle que la liaison SOAP. L'acquisition d'un message de protocole SAML étant réussie, l'artifice est éliminé, et le traitement de l'échange principal de protocole SAML reprend (ou se termine, si le message est une réponse).

Produire et livrer un artifice, joint à l'étape de résolution ultérieure, constitue la moitié de l'échange global de protocole SAML. Cette liaison peut être utilisée pour livrer l'une ou l'autre ou les deux moitiés d'un échange de protocole SAML. Une liaison composable avec lui, telle qu'une liaison HTTP Redirect (voir au § 10.2.4) ou POST (voir au § 10.2.5), peut être utilisée pour porter l'autre moitié de l'échange. La séquence suivante suppose que la liaison artifice est utilisée pour les deux moitiés. Voir la Figure 10-4 ci-dessous qui illustre les messages échangés.

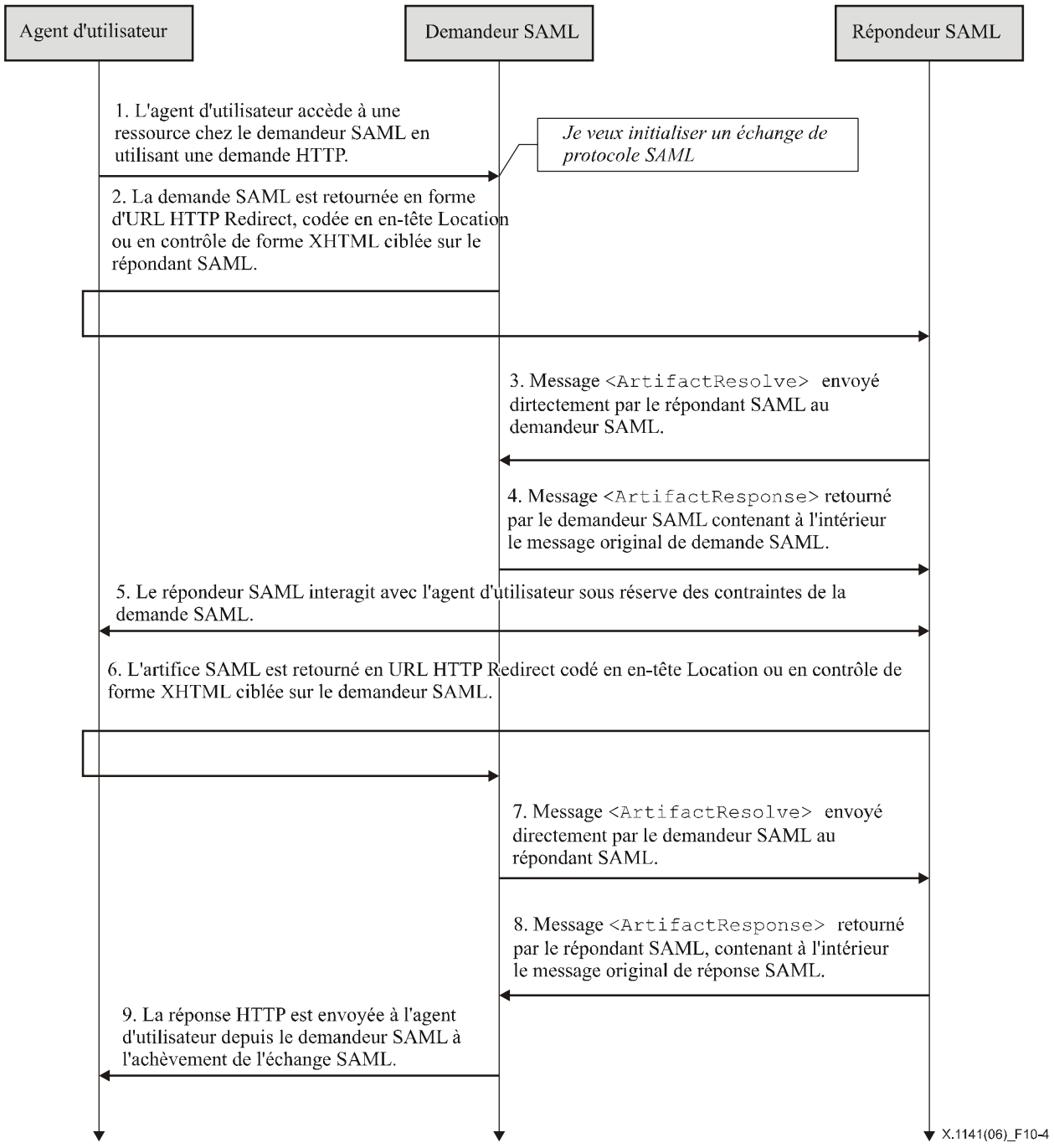

Figure 10-4/X.1141 – Échange de message HTTP Artifact

1) Au départ, l'agent d'utilisateur fait une demande HTTP arbitraire à une entité système. Dans le cours du traitement de la demande, l'entité système décide d'initialiser un échange de protocole SAML.

2) L'entité système qui agit comme demandeur SAML répond à une demande HTTP de l'agent d'utilisateur en retournant un artifice représentant une demande SAML.

   - S'il est codé en URL, l'artifice est retourné codé dans l'en-tête Location de la réponse HTTP, et l'état HTTP doit être 303 ou 302. Le demandeur SAML peut inclure une présentation et un contenu supplémentaires dans la réponse HTTP pour faciliter la transmission du message par l'agent d'utilisateur, comme défini dans la RFC 2616 de l'IETF. L'agent d'utilisateur livre l'artifice en produisant une demande HTTP GET au répondant SAML.

- S'il est codé en forme, l'artifice est alors retourné dans un document XHTML contenant la forme et le contenu définis au paragraphe 10.2.6.3.3. L'agent d'utilisateur livre l'artifice en produisant une demande HTTP POST au répondant SAML.

3) Le répondant SAML détermine qui est le demandeur SAML en examinant l'artifice (le processus exact dépend du type d'artifice), et produit une demande `<samlp:ArtifactResolve>` qui contient l'artifice, au demandeur SAML en utilisant une liaison SAML directe, renversant temporairement les rôles.

4) En supposant que les conditions nécessaires sont satisfaites, le demandeur SAML retourne une `<samlp:ArtifactResponse>` contenant le message de demande SAML d'origine qu'il souhaite que traite le répondant SAML.

5) En général, le répondant SAML peut répondre à la demande SAML en retournant immédiatement un artifice SAML ou peut retourner un contenu arbitraire pour faciliter une interaction ultérieure avec l'agent d'utilisateur nécessaire pour satisfaire la demande. Des protocoles et profils spécifiques peuvent inclure des mécanismes pour indiquer le niveau de volonté du demandeur de permettre ce type d'interaction (par exemple, l'attribut `IsPassive` dans `<samlp:AuthnRequest>`).

6) Finalement, le répondant devrait retourner un artifice SAML à l'agent d'utilisateur pour qu'il soit retourné au demandeur SAML. L'artifice de réponse SAML est retourné de la même façon que décrite pour l'artifice de demande SAML de l'étape 2.

7) Le demandeur SAML détermine qui est le répondant SAML en examinant l'artifice, et produit une demande `<samlp:ArtifactResolve>` qui contient l'artifice au répondant SAML en utilisant une liaison SAML directe, comme à l'étape 3.

   NOTE (informative) – PE31 (voir OASIS PE:2006) suggère de remplacer la dernière phrase de l'étape 6 par:

   Le demandeur SAML détermine qui est le répondant SAML en examinant l'artifice, et produit une demande `<samlp:ArtifactResolve>` qui contient l'artifice au répondant SAML en utilisant une liaison SAML synchrone, comme à l'étape 3.

8) En supposant que les conditions nécessaires sont satisfaites, le répondant SAML retourne un `<samlp:ArtifactResponse>` contenant le message de réponse SAML qu'il souhaite que traite le demandeur, comme à l'étape 4.

9) A réception de la réponse SAML, le demandeur SAML retourne une réponse HTTP arbitraire à l'agent d'utilisateur.

10.2.6.5.1 HTTP et considérations de mise en mémoire cache

Les mandataires HTTP et l'intermédiaure d'agent d'utilisateur ne devraient pas mettre en mémoire cache les artifices SAML. Pour s'en assurer, les règles suivantes devrait être suivies.

En retournant les artifices SAML utilisant HTTP 1.1, les répondants HTTP devraient:

- inclure un champ d'en-tête `Cache-Control` réglé à "`no-cache`, `no-store`".
- inclure un champ d'en-tête `Pragma` réglé à "`no-cache`".

Il n'y a aucune autre restriction à l'utilisation des en-têtes HTTP.

10.2.6.5.2 Considérations sur la sécurité

Cette liaison utilise une combinaison de transmission indirecte de référence de message suivie par un échange direct pour retourner le message réel. Il en résulte que la référence de message (l'artifice) n'a pas besoin d'être elle-même authentifiée ou protégée en intégrité, mais l'échange demande/réponse en retour qui retourne le message réel peut être mutuellement authentifié et protégé en intégrité, selon l'environnement d'utilisation.

Si le message de protocole SAML réel est destiné à un receveur spécifique, le producteur de l'artifice doit alors authentifier l'envoyeur du message `<samlp:ArtifactResolve>` suivant avant de retourner le message réel.

La transmission d'un artifice de et vers l'agent d'utilisateur devrait être protégée par la confidentialité; ou TLS 1.0 devrait être utilisé. L'échange demande/réponse en retour qui retourne le message réel peut être protégé, selon l'environnement d'utilisation.

En général, cette liaison s'appuie sur l'artifice comme référence à court terme difficile à falsifier et applique d'autres mesures de sécurité à la demande/réponse en retour qui retourne le message réel. Tous les artifices doivent avoir une sémantique d'utilisation unique mise en application par le producteur de l'artifice.

De plus, il est recommandé que les receveurs d'artifice mettent aussi en application une sémantique d'utilisation unique aux valeurs d'artifices qu'ils reçoivent, pour empêcher un attaquant d'interférer avec la résolution d'un artifice par un agent d'utilisateur et de le resoumettre ensuite au receveur de l'artifice. Si une tentative de résolution d'un artifice ne

s'achève pas avec succès, l'artifice devrait être placé dans une liste d'artifices bloqués pour une durée qui excède la période raisonnable d'acceptation pendant laquelle le producteur de l'artifice devrait résoudre l'artifice.

Il n'y a pas de mécanisme défini pour protéger l'intégrité de la relation entre l'artifice et la valeur "RelayState", s'il en est. C'est-à-dire qu'un attaquant a la possibilité de recombiner une paire de réponses HTTP valides en commutant les valeurs "RelayState" associées à chaque artifice. Il en résulte que le producteur/consommateur d'informations "RelayState" doit veiller à ne pas associer d'informations d'état sensibles à la valeur "RelayState" sans prendre de précautions supplémentaires (telles que de se fonder sur les informations du message de protocole SAML restituées via l'artifice).

10.2.6.6 Rapport d'erreurs

Un répondant SAML qui refuse d'effectuer un échange de messages avec le demandeur SAML devrait retourner un message de réponse avec une valeur `<samlp:StatusCode>` de second niveau `urn:oasis:names:tc:SAML:2.0:status:RequestDenied`.

Les interactions HTTP durant les échanges de messages ne doivent pas utiliser des codes d'état d'erreur pour indiquer des échecs du traitement SAML, car l'agent d'utilisateur n'est pas partie prenante à part entière à l'échange de protocole SAML.

Si le producteur d'un artifice reçoit un message `<samlp:ArtifactResolve>` qu'il peut comprendre, il doit retourner un `<samlp:ArtifactResponse>` avec une valeur `<samlp:StatusCode>` de `urn:oasis:names:tc:SAML:2.0:status:Success`, même s'il ne retourne pas le message correspondant (par exemple, parce que le demandeur de l'artifice n'est pas autorisé à recevoir le message ou que l'artifice n'est plus valide).

10.2.6.7 Considérations sur les métadonnées

La prise en charge des liaisons HTTP Artifact devrait être reflétée par l'indication des points d'extrémité d'URL auxquels les demandes et réponses pour un protocole ou profil particulier devraient être envoyées. Un seul point d'extrémité ou des points d'extrémité distincts pour la demande et la réponse peuvent être fournis. Un ou plusieurs points d'extrémité indexés devraient aussi être décrits pour le traitement des messages `<samlp:ArtifactResolve>`.

10.2.6.8 Exemple d'échange de messages SAML utilisant HTTP Artifact

Dans cet exemple, une paire de messages `<LogoutRequest>` et `<LogoutResponse>` est échangée en utilisant la liaison HTTP Artifact, la résolution de l'artifice ayant lieu en utilisant la liaison SOAP attachée à HTTP.

Tout d'abord figurent ici les messages de protocole SAML réels qui sont échangés:

```
<samlp:LogoutRequest xmlns:samlp="urn:oasis:names:tc:SAML:2.0:protocol"
xmlns="urn:oasis:names:tc:SAML:2.0:assertion"
    ID="d2b7c388cec36fa7c39c28fd298644a8" IssueInstant="2004-01-
21T19:00:49Z" Version="2.0">
    <Issuer>https://IdentityProvider.com/SAML</Issuer>
    <NameID Format="urn:oasis:names:tc:SAML:2.0:nameid-
format:persistent">005a06e0-ad82-110d-a556-004005b13a2b</NameID>
    <samlp:SessionIndex>1</samlp:SessionIndex>
</samlp:LogoutRequest>

<samlp:LogoutResponse xmlns:samlp="urn:oasis:names:tc:SAML:2.0:protocol"
xmlns="urn:oasis:names:tc:SAML:2.0:assertion"
    ID="b0730d21b628110d8b7e004005b13a2b"
InResponseTo="d2b7c388cec36fa7c39c28fd298644a8"
    IssueInstant="2004-01-21T19:00:49Z" Version="2.0">
    <Issuer>https://ServiceProvider.com/SAML</Issuer>
    <samlp:Status>
        <samlp:StatusCode
Value="urn:oasis:names:tc:SAML:2.0:status:Success"/>
    </samlp:Status>
</samlp:LogoutResponse>
```

La demande HTTP initiale de l'agent d'utilisateur à l'étape 1 n'est pas définie par cette liaison. Pour initier l'échange de protocole de terminaison de session, le demandeur SAML retourne la réponse HTTP suivante, qui contient un artifice SAML. Les renvois à la ligne dans l'en-tête HTTP Location ci-dessous résultent du formatage de document, et il n'y a pas de renvois à la ligne dans la valeur d'en-tête réelle.

```
HTTP/1.1 302 Object Moved
Date: 21 Jan 2004 07:00:49 GMT
Location:
https://ServiceProvider.com/SAML/SLO/Browser?SAMLart=AAQAADWNEw5VT47wcO4zX%
2FiEzMmFQvGknDfws2ZtqSGdkNSbsW1cmVR0bzU%3D&RelayState=0043bfc1bc45110dae170
04005b13a2b
Content-Type: text/html; charset=iso-8859-1
```

Le répondant SAML résoud ensuite l'artifice qu'il a reçu dans la demande SAML réelle en utilisant le protocole de résolution d'artifice et la liaison SOAP dans les étapes 3 et 4, comme suit:

Etape 3:

```
POST /SAML/Artifact/Resolve HTTP/1.1
Host: IdentityProvider.com
Content-Type: text/xml
Content-Length: nnn
SOAPAction: http://www.oasis-open.org/committees/security
<SOAP-ENV:Envelope
    xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
    <SOAP-ENV:Body>
        <samlp:ArtifactResolve
             xmlns:samlp="urn:oasis:names:tc:SAML:2.0:protocol"
             xmlns="urn:oasis:names:tc:SAML:2.0:assertion"
             ID="_6c3a4f8b9c2d" Version="2.0"
             IssueInstant="2004-01-21T19:00:49Z">
             <Issuer>https://ServiceProvider.com/SAML</Issuer>
             <Artifact>
             AAQAADWNEw5VT47wcO4zX/iEzMmFQvGknDfws2ZtqSGdkNSbsW1cmVR0bzU=
             </Artifact>
        </samlp:ArtifactResolve>
    </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Etape 4:

```
HTTP/1.1 200 OK
Date: 21 Jan 2004 07:00:49 GMT
Content-Type: text/xml
Content-Length: nnnn

<SOAP-ENV:Envelope
    xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
    <SOAP-ENV:Body>
        <samlp:ArtifactResponse
             xmlns:samlp="urn:oasis:names:tc:SAML:2.0:protocol"
             xmlns="urn:oasis:names:tc:SAML:2.0:assertion"
             ID="_FQvGknDfws2Z" Version="2.0"
             InResponseTo="_6c3a4f8b9c2d"
             IssueInstant="2004-01-21T19:00:49Z">
             <Issuer>https://IdentityProvider.com/SAML</Issuer>
             <samlp:Status>
                    <samlp:StatusCode
             Value="urn:oasis:names:tc:SAML:2.0:status:Success"/>
             </samlp:Status>
             <samlp:LogoutRequest ID="d2b7c388cec36fa7c39c28fd298644a8"
                    IssueInstant="2004-01-21T19:00:49Z"
                    Version="2.0">
                    <Issuer>https://IdentityProvider.com/SAML</Issuer>
                    <NameID Format="urn:oasis:names:tc:SAML:2.0:nameid-
format:persistent">005a06e0-ad82-110d-a556-004005b13a2b</NameID>
                    <samlp:SessionIndex>1</samlp:SessionIndex>
             </samlp:LogoutRequest>
        </samlp:ArtifactResponse>
    </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Après que toutes les interactions non spécifiées ont pu avoir lieu, le répondant SAML retourne un second artifice SAML dans sa réponse HTTP à l'étape 6:

```
HTTP/1.1 302 Object Moved
Date: 21 Jan 2004 07:05:49 GMT
Location:
https://IdentityProvider.com/SAML/SLO/Response?SAMLart=AAQAAFGIZXv5%2BQaBaE
5qYurHWJO1nAgLAsqfnyiDHIggbFU0mlSGFTyQiPc%3D&RelayState=0043bfc1bc45110dae1
7004005b13a2b
Content-Type: text/html; charset=iso-8859-1
```

Le répondant SAML résout ensuite l'artifice qu'il a reçu dans la demande SAML réelle en utilisant le protocole de résolution d'artifice et la liaison SOAP dans les étapes 7 et 8, comme suit:

Etape 7:

```
POST /SAML/Artifact/Resolve HTTP/1.1
Host: ServiceProvider.com
Content-Type: text/xml
Content-Length: nnn
SOAPAction: http://www.oasis-open.org/committees/security
<SOAP-ENV:Envelope
    xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
    <SOAP-ENV:Body>
        <samlp:ArtifactResolve
             xmlns:samlp="urn:oasis:names:tc:SAML:2.0:protocol"
             xmlns="urn:oasis:names:tc:SAML:2.0:assertion"
             ID="_ec36fa7c39" Version="2.0"
             IssueInstant="2004-01-21T19:05:49Z">
             <Issuer>https://IdentityProvider.com/SAML</Issuer>
             <Artifact>
             AAQAAFGIZXv5+QaBaE5qYurHWJO1nAgLAsqfnyiDHIggbFU0mlSGFTyQiPc=
             </Artifact>
        </samlp:ArtifactResolve>
    </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Etape 8:

```
HTTP/1.1 200 OK
Date: 21 Jan 2004 07:05:49 GMT
Content-Type: text/xml
Content-Length: nnnn

<SOAP-ENV:Envelope
    xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
    <SOAP-ENV:Body>
        <samlp:ArtifactResponse
             xmlns:samlp="urn:oasis:names:tc:SAML:2.0:protocol"
             xmlns="urn:oasis:names:tc:SAML:2.0:assertion"
             ID="_FQvGknDfws2Z" Version="2.0"
             InResponseTo="_ec36fa7c39"
             IssueInstant="2004-01-21T19:05:49Z">
             <Issuer>https://ServiceProvider.com/SAML</Issuer>
             <samlp:Status>
                    <samlp:StatusCode
             Value="urn:oasis:names:tc:SAML:2.0:status:Success"/>
             </samlp:Status>
             <samlp:LogoutResponse ID="_b0730d21b628110d8b7e004005b13a2b"
                    InResponseTo="_d2b7c388cec36fa7c39c28fd298644a8"
                    IssueInstant="2004-01-21T19:05:49Z"
                    Version="2.0">
                    <Issuer>https://ServiceProvider.com/SAML</Issuer>
                    <samlp:Status>
                           <samlp:StatusCode
             Value="urn:oasis:names:tc:SAML:2.0:status:Success"/>
                    </samlp:Status>
             </samlp:LogoutResponse>
        </samlp:ArtifactResponse>
    </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

10.2.7 Liaison d'URI SAML

Les URI sont un moyen indépendant du protocole de se référer à une ressource. Cette liaison n'est pas une liaison générale de demande/réponse SAML, mais plutôt un support de l'encapsulation d'un message `<samlp:AssertionIDRequest>` avec un seul `<saml:AssertionIDRef>` dans la résolution d'un URI. Le résultat d'une demande réussie est un élément `<saml:Assertion>` SAML (mais pas une réponse SAML complète).

Comme pour SOAP, la résolution d'URI peut survenir sur plusieurs transports sous-jacents. Cette liaison a des aspects indépendants du transport, mais appelle aussi à l'utilisation de HTTP avec TLS 1.0 en tant que de besoin (implémentation obligatoire).

NOTE (informative) – PE24 (voir OASIS PE:2006) suggère de remplacer l'alinéa ci-dessus par ce qui suit:

Comme pour SOAP, la résolution d'URI peut survenir sur plusieurs transports sous-jacents. Cette liaison a des aspects indépendants du protocole, mais appelle aussi l'implémentation obligatoire des URI HTTP.

10.2.7.1 Informations requises

Identification: `urn:oasis:names:tc:SAML:2.0:bindings:URI`

Informations de contact: security-services-comment@lists.oasis-open.org

Description: donnée ci-dessous

Mises à jour: aucune

10.2.7.2 Aspects indépendants du protocole de la liaison d'URI SAML

Les paragraphes suivants définissent les aspects de la liaison d'URI SAML qui sont indépendants du protocole de transport sous-jacent du processus de résolution d'URI.

Une référence d'URI SAML identifie une assertion SAML spécifique. Le résultat de la résolution de l'URI doit être un message contenant l'assertion, ou une erreur spécifique du transport. Le format spécifique du message dépend du protocole de transport sous-jacent. Si le protocole de transport permet la description du contenu retourné, comme avec HTTP 1.1, l'assertion peut alors être codée dans tout format permis. Sinon, l'assertion doit être retournée sous une forme qui puisse être interprétée de façon non ambiguë ou transformée en une sérialisation XML de l'assertion.

Si la même référence d'URI est résolue à l'avenir, la même assertion SAML, ou une erreur, doit être retournée. C'est-à-dire que la référence peut être persistante mais doit faire référence de manière cohérente à la même assertion, s'il en est.

10.2.7.3 Considérations sur la sécurité

L'utilisation indirecte d'une assertion SAML présente des dangers si la liaison de la référence au résultat n'est pas sécurisée. Les menaces particulières et leur sévérité dépendent de l'utilisation de l'assertion. En général, le résultat de la résolution d'une référence d'URI à une assertion SAML devrait n'être de confiance que si le demandeur peut être certain de l'identité du répondant et que le contenu n'a pas été modifié dans le transit.

Il n'est souvent pas suffisant que l'assertion soit elle-même signée, parce que les références d'URI sont par nature assez opaques au demandeur. Le demandeur devrait avoir des moyens indépendants de s'assurer que l'assertion retournée est réellement celle qui est représentée par l'URI; cela se fait à la fois par l'authentification du répondant et en s'appuyant sur l'intégrité de la réponse.

10.2.7.4 Encapsulation MIME

Pour les protocoles de résolution qui prennent en charge MIME comme description de contenu et mécanisme de paquetage, l'assertion qui en résulte devrait être retournée comme une entité MIME de type `application/samlassertion+xml`, comme défini dans l'Appendice II.

10.2.7.5 Utilisation des URI HTTP

Une autorité SAML qui revendique la conformité à la liaison d'URI SAML doit mettre en œuvre la prise en charge de HTTP. Le présent paragraphe décrit certaines spécificités de l'utilisation des URI HTTP, y compris la syntaxe d'URI, les en-têtes HTTP, et le rapport d'erreurs.

10.2.7.5.1 Syntaxe d'URI

En général, il n'y a pas de restriction sur la syntaxe permise d'une référence d'URI SAML pour autant que l'autorité SAML responsable de la référence crée le message qui la contient. Cependant, les autorités doivent prendre en charge un point d'extrémité d'URL auquel une demande HTTP peut être envoyée avec un seul `ID` (identifiant) désigné de

paramètre de chaîne d'interrogation. Il ne doit y avoir aucune chaîne d'interrogation dans l'URL de point d'extrémité lui-même indépendamment de ce paramètre.

Par exemple, si le point d'extrémité référencé à une autorité est "https://saml.example.edu/assertions", une demande d'assertion avec un `ID` de `abcde` peut être envoyée à:

```
https://saml.example.edu/assertions?ID=abcde
```

L'utilisation de caractères génériques n'est pas admise pour de telles interrogations d'ID.

NOTE (informative) – PE31 (voir OASIS PE:2006) suggère de remplacer le texte ci-dessus par:

Noter que la syntaxe d'URI n'accepte pas l'utilisation de caractères génériques dans de telles interrogations.

10.2.7.5.2 HTTP et considérations sur la mise en mémoire cache

Les mandataires HTTP ne doivent pas mettre en mémoire cache les assertions SAML. Pour s'en assurer, les règles suivantes devrait être suivies.

En retournant les assertions SAML à l'aide de HTTP 1.1, les répondants HTTP devraient:

- inclure un champ d'en-tête `Cache-Control` réglé à "`no-cache, no-store`".
- inclure un champ d'en-tête `Pragma` réglé à to "`no-cache`".

10.2.7.5.3 Considérations sur la sécurité

La RFC 2617 de l'IETF décrit des attaques possibles dans l'environnement HTTP lorsque les schémas d'authentification de base ou de résumé de message sont utilisés.

L'utilisation de TLS 1.0 est fortement recommandée comme moyen d'authentification, de protection de l'intégrité, et de confidentialité.

10.2.7.5.4 Rapport d'erreurs

Comme pour un échange de protocole HTTP, le code d'état HTTP approprié devrait être utilisé pour indiquer le résultat d'une demande. Par exemple, un répondant SAML qui refuse d'effectuer un échange de messages avec le demandeur SAML devrait retourner une réponse "`403 Forbidden` (Interdit)". Si l'assertion spécifiée est inconnue du répondant, une réponse "`404 Not Found` (Pas trouvé)" devrait être retournée. Dans ces cas, le contenu du corps HTTP n'est pas significatif.

10.2.7.5.5 Considérations sur les métadonnées

La prise en charge de la liaison d'URI sur HTTP devrait être reflétée en indiquant un point d'extrémité d'URL auquel sera envoyée la demande d'assertions arbitraires.

10.2.7.5.6 Exemple d'échange de messages SAML utilisant un URI HTTP

Ci-après figure un exemple d'une demande d'assertion.

```
GET /SamlService?ID=abcde HTTP/1.1
Host: www.example.com
```

Ci-après figure un exemple de la réponse correspondante, qui fournit l'assertion demandée.

```
HTTP/1.1 200 OK
Content-Type: application/samlassertion+xml
Cache-Control: no-cache, no-store
Pragma: no-cache
Content-Length: nnnn

<saml:Assertion ID="abcde" ...>
...
</saml:Assertion>
```

11 Profils pour SAML

Le présent paragraphe spécifie les profils qui définissent l'utilisation des assertions SAML et messages de demande-réponse dans les protocoles et cadres de communications, ainsi que les profils qui définissent la syntaxe de valeur d'attribut SAML et les conventions de dénomination.

11.1 Concepts de profil

Un type de profil SAML esquisse un ensemble de règles qui décrivent comment enchasser les assertions SAML dans un cadre de travail ou un protocle et comment les en extraire. Un tel profil décrit comment les assertions SAML sont enchassées dans, ou combinées avec, d'autres objets (par exemple, fichiers de divers types, ou unités de données de protocole de protocoles de communication) par une partie d'origine, communiquées de la partie d'origine à une partie receveuse, et ensuite traitées à destination. Un ensemble particulier de règles pour enchasser les assertions SAML dans une classe spécifiée d'objets `<FOO>` et les en extraire est appelé un *profil<FOO> de SAML*.

Par exemple, un profil SOAP de SAML décrit comment les assertions SAML peuvent être ajoutées aux messages SOAP, comment les en-têtes SOAP sont affectés par les assertions SAML, et comment les états d'erreurs en rapport avec SAML devraient être reflétés dans les messages SOAP.

Un autre type de profil SAML définit un ensemble de contraintes sur l'utilisation d'une capacité générale de protocole ou d'assertion SAML pour un environnement particulier ou contexte d'utilisation. Les profils de cette nature peuvent restreindre le caractère facultatif, exiger l'utilisation de fonctionnalités SAML spécifiques (par exemple, des attributs, des conditions, ou des liaisons), et sous d'autres aspects, définir les règles de traitement à suivre pour les acteurs du profil.

Un exemple particulier de ce dernier cas est celui qui vise les attributs SAML. L'élément `<Attribute>` SAML fournit une grande souplesse à la dénomination des attributs, à la syntaxe de valeur, et à l'inclusion dans la bande de métadonnées par l'utilisation des attributs XML. L'interopérabilité est réalisée en restreignant cette souplesse quand elle est garantie par l'adhésion aux profils qui définissent comment utiliser ces éléments avec une spécificité plus importante que les règles générales définies au § 8.

Les profils d'attribut fournissent les définitions nécessaires pour restreindre l'expression des attributs SAML lorsqu'on traite de types particuliers d'informations d'attribut ou lors d'interactions avec des systèmes externes ou autres standards ouverts qui ont des exigences plus strictes.

La présente Recommandation a l'intention de spécifier un ensemble choisi de profils de diverses sortes avec un niveau de détail suffisant pour assurer l'interfonctionnement des produits implémenté de façon indépendante.

11.2 Spécification de profils supplémentaires

La présente Recommandation définit un ensemble choisi de profils, mais d'autres pourront être développés à l'avenir. Les paragraphes suivants proposent des lignes directrices pour la spécification de profils.

11.2.1 Lignes directrices pour spécifier des profils

Le présent paragraphe fournit une liste de contrôle des questions qui doivent être traitées par chaque profil.

1) Spécifier un URI qui identifie de façon univoque le profil, les informations de contact postal ou électronique sur l'auteur, et fournisse la référence des profils précédemment définis que le nouveau profil met à jour ou rend obsolètes.
2) Descrire l'ensemble des interactions entre parties impliquées dans le profil. Toutes les restrictions sur les applications utilisées par chaque partie et les protocoles impliqués dans chaque interaction doivent être explicitement mentionnées.
3) Identifier les parties impliquées dans chaque interaction, y compris comment chaque partie est impliquée et si des intermédiaires peuvent l'être.
4) Spécifier la méthode d'authentification des parties impliquées dans chaque interaction, y compris si l'authentification est exigée et les types d'authentification acceptables.
5) Identifier le niveau de prise en charge de l'intégrité du message, y compris les mécanismes utilisés pour assurer l'intégrité du message.
6) Identifier le niveau de prise en charge de la confidentialité, y compris si un tiers peut voir les contenus des messages et assertions SAML, si le profil exige la confidentialité, et les mécanismes recommandés pour réaliser la confidentialité.
7) Identifier les états d'erreurs, y compris les états d'erreur chez chaque participant, particulièrement ceux qui reçoivent et traitent les assertions ou messages SAML.
8) Identifier les considérations de sécurité, y compris l'analyse des menaces et la description des contre-mesures.
9) Identifier les identifiants de méthode de confirmation SAML définis et/ou utilisés par le profil.
10) Identifier les métadonnées SAML pertinentes définies et/ou utilisées par le profil.

11.2.2 Lignes directrices pour spécifier des profils d'attribut

Le présent paragraphe fournit une liste de contrôle des éléments qui doivent en particulier être traités par les profils d'attribut.

1) Spécifier un URI qui identifie de façon univoque le profil, les informations de contact postal ou électronique sur l'auteur, et fournisse la référence des profils précédemment définis que le nouveau profil met à jour ou rend obsolètes.
2) La syntaxe et les restrictions sur les valeurs acceptables des attributs `NameFormat` et `Name` des éléments `<Attribute>` SAML.
3) Tous les attributs XML supplémentaires d'espace de nom qualifié définis par le profil qui peuvent être utilisés dans les éléments `<Attribute>` SAML.
4) Les règles de détermination de l'égalité des éléments `<Attribute>` SAML comme défini par le profil, à utiliser lors du traitement des attributs, interrogations, etc.
5) La syntaxe et les restrictions sur les valeurs acceptables dans l'élément `<AttributeValue>` SAML, y compris si l'attribut XML `xsi:type` peut ou devrait être utilisé.

11.3 Identifiants de méthode de confirmation

Le paragraphe 8 définit l'élément `<SubjectConfirmation>` comme Method plus `<SubjectConfirmationData>` facultatif. L'élément `<SubjectConfirmation>` devrait être utilisé par le consommateur d'assertions pour confirmer que la demande ou message est venu d'une entité système associée au sujet de l'assertion, dans le contexte d'un profil particulier.

L'attribut `Method` indique la méthode spécifique que le consommateur d'assertions devrait utiliser pour faire cette détermination. Cela peut avoir ou n'avoir pas de relation avec une authentification effectuée précédemment. A la différence du contexte d'authentification, la méthode de confirmation du sujet sera souvent accompagnée d'informations supplémentaires, telles qu'un certificat ou une clé, dans l'élément `<SubjectConfirmationData>` qui va permettre au consommateur d'assertions d'effectuer la vérification nécessaire. Un ensemble commun d'attributs est aussi défini et peut être utilisé pour restreindre les conditions dans lesquelles la vérification peut avoir lieu.

On prévoit que les profils définiront et utiliseront plusieurs valeurs différentes pour `<ConfirmationMethod>`, chacune correspondant à un scénario d'utilisation différent de SAML. Les méthodes suivantes sont définies pour l'utilisation par des profils définis dans la présente Recommandation et d'autres profils qui les trouvent utiles.

11.3.1 Détenteur de clé

URI: `urn:oasis:names:tc:SAML:2.0:cm:holder-of-key`

Un ou plusieurs éléments `<ds:KeyInfo>` doivent être présents au sein de l'élément `<SubjectConfirmationData>`. Un attribut `xsi:type` peut être présent dans l'élément `<SubjectConfirmationData>` et, s'il est présent, doit être réglé à **saml:KeyInfoConfirmationDataType** (l'espace de nom prefix est arbitraire mais doit référencer l'espace de nom d'assertion SAML).

Comme décrit dans le document Signature du W3C, chaque élément `<ds:KeyInfo>` détient une clé ou des informations qui permettent à une application d'obtenir une clé. Le détenteur d'une clé spécifiée est considéré comme le sujet de l'assertion par le producteur d'assertions.

Conformément au document Signature du W3C, chaque élément `<ds:KeyInfo>` doit identifier une seule clé cryptographique. Plusieurs clés peuvent être identifiées avec des éléments `<ds:KeyInfo>` séparés, comme lorsque différentes clés de confirmation sont nécessaires pour différents consommateurs d'assertions.

Exemple: le détenteur de la clé nommée "By-Tor" ou le détenteur de la clé nommée "Snow Dog" peut se confirmer lui-même comme le sujet.

```
<SubjectConfirmation Method="urn:oasis:names:tc:SAML:2.0:cm:holder-of-key">
       <SubjectConfirmationData xsi:type="saml:KeyInfoConfirmationDataType">
              <ds:KeyInfo>
                     <ds:KeyName>By-Tor</ds:KeyName>
              </ds:KeyInfo>
              <ds:KeyInfo>
                     <ds:KeyName>Snow Dog</ds:KeyName>
              </ds:KeyInfo>
       </SubjectConfirmationData>
</SubjectConfirmation>
```

11.3.2 Garantie de l'envoyeur

URI: urn:oasis:names:tc:SAML:2.0:cm:sender-vouches

Indique qu'aucune autre information n'est disponible sur le contexte d'utilisation de l'assertion. Le consommateur d'assertions devrait utiliser d'autres moyens pour déterminer s'il devrait traiter l'assertion, sous réserve des restrictions facultatives sur la confirmation qui utilisent les attributs qui peuvent être présentes dans l'élément <SubjectConfirmationData>.

11.3.3 Titulaire

URI: urn:oasis:names:tc:SAML:2.0:cm:bearer

Le sujet de l'assertion est le tutulaire de l'assertion, sous réserve des restrictions facultatives sur la confirmation qui utilisent les attributs pouvant être présentes dans l'élément <SubjectConfirmationData>, comme défini au § 8.

Exemple: le titulaire de l'assertion peut se confirmer lui-même comme le sujet, pourvu que l'assertion soit délivrée dans un message envoyé à "https://www.serviceprovider.com/saml/consumer" avant 13:27 GMT le 19 mars 2004, en réponse à une demande avec l'ID "_1234567890".

```
<SubjectConfirmation Method="urn:oasis:names:tc:SAML:2.0:cm:bearer">
       <SubjectConfirmationData InResponseTo="_1234567890"
              Recipient="https://www.serviceprovider.com/saml/consumer"
              NotOnOrAfter="2004-03-19T13:27:00Z"
       </SubjectConfirmationData>
</SubjectConfirmation>
```

11.4 Profils SSO de SAML

Un ensemble de profils est défini pour prendre en charge la signature unique (SSO, *single sign-on*) des navigateurs et autres appareils clients.

- Il est défini un profil, fondé sur un navigateur de la toile, du protocole de demande d'authentification du § 8 pour la prise en charge de la signature unique sur la toile.
- Un profil supplémentaire SSO de la toile est défini pour la prise en charge des clients améliorés.
- Un profil des protocoles de fermeture de session unique et de gestion d'identifiant de nom du § 8 est défini à la fois sur les liaisons de canal frontal (navigateur) et de canal arrière.
- Un profil supplémentaire est défini pour la découverte de fournisseur d'identité en utilisant des mouchards *(cookies)*.

11.4.1 Profil SSO de navigateur de la toile

Dans le scénario pris en charge par le profil SSO de navigateur de la toile, un utilisateur de la toile accède à une ressource chez un fournisseur de service, ou accède à un fournisseur d'identité tel que le fournisseur de service et la ressource désirée soient compris ou implicites. L'utilisateur de la toile s'authentifie (ou s'est déjà authentifié) auprès du fournisseur d'identité, qui produit alors une assertion d'authentification (éventuellement avec un apport de la part du fournisseur de service) et le fournisseur de service consomme l'assertion pour établir un contexte de sécurité pour l'utilisateur de la toile. Durant ce processus, un identifiant de nom peut aussi être établi entre les fournisseurs pour le principal, sous réserve des paramètres de l'interaction et du consentement des parties.

Pour implémenter ce scénario, un profil du protocole de demande d'authentification SAML est utilisé, conjointement avec les liaisons HTTP Redirect, HTTP POST et HTTP Artifact.

On suppose que l'usager utilise un navigateur commercial standard et peut s'authentifier au fournisseur d'identité par des moyens qui sortent du domaine d'application de SAML.

11.4.1.1 Informations requises

Identification: urn:oasis:names:tc:SAML:2.0:profiles:SSO:browser

Informations de contact: security-services-comment@lists.oasis-open.org

Identifiants de méthode de confirmation SAML: l'identifiant de méthode de confirmation "bearer" de SAML V2.0, urn:oasis:names:tc:SAML:2.0:cm:bearer, est utilisé par ce profil.

Description: donnée ci-dessous.

Mises à jour: aucune.

11.4.1.2 Aperçu général du profil

La Figure 11-1 illustre le schéma de base de la réalisation de SSO. Les étapes suivantes sont décrites par le profil. Dans une étape individuelle, il peut y avoir un ou plusieurs échanges de messages réels selon la liaison utilisée pour cette étape et d'autres comportements dépendants de l'implémentation.

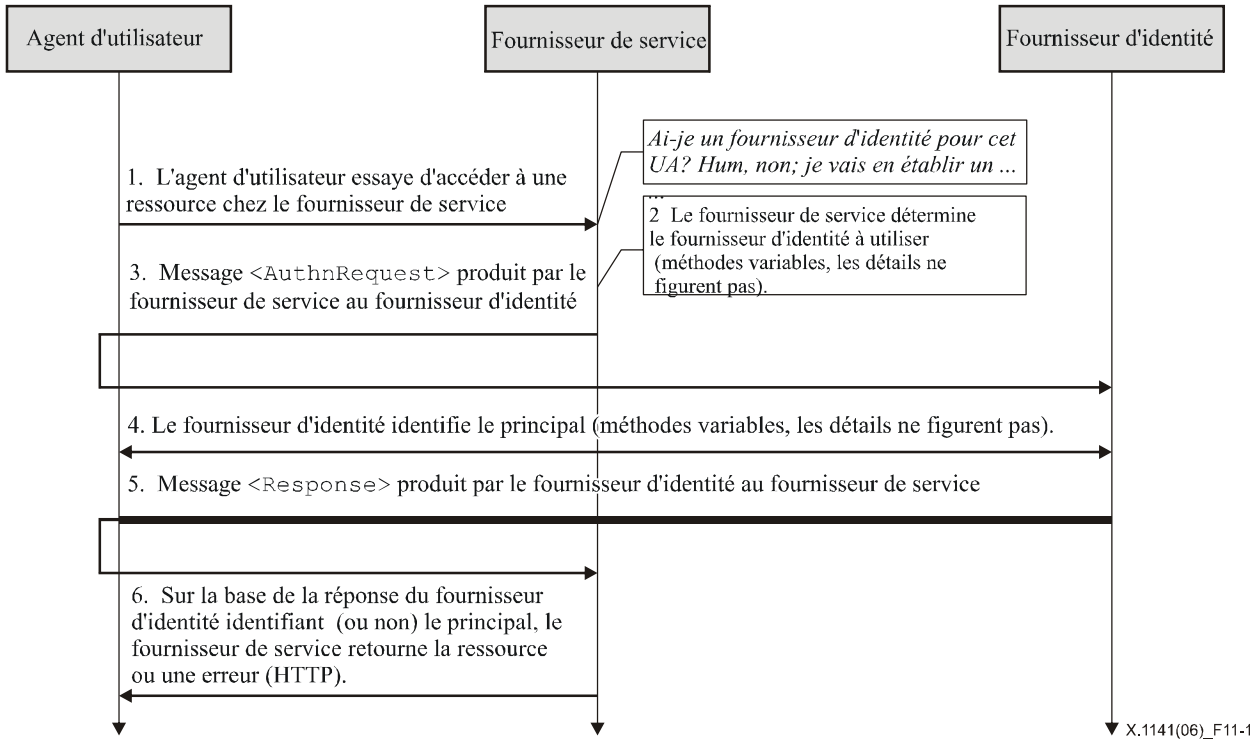

Figure 11-1/X.1141 – Schéma de base pour la réalisation de SSO

1) Demande HTTP au fournisseur de service

A l'étape 1, le principal, via un agent d'utilisateur HTTP, fait une demande HTTP de ressource sécurisée au fournisseur de service sans contexte de sécurité.

2) Le fournisseur de service détermine un fournisseur d'identité

A l'étape 2, le fournisseur de service obtient la localisation d'un point d'extrémité chez un fournisseur d'identité pour le protocole de demande d'authentification qui prend en charge sa liaison préférée. Le moyen par lequel ceci est réalisé dépend de l'implémentation. Le fournisseur de service peut utiliser le profil de découverte de fournisseur d'identité SAML décrit au § 8.7.4.

3) <AuthnRequest> est produit par le fournisseur de service au fournisseur d'identité

A l'étape 3, le fournisseur de service produit un message `<AuthnRequest>` à délivrer par l'agent d'utilisateur au fournisseur d'identité. La liaison HTTP Redirect, HTTP POST, ou HTTP Artifact peut être utilisée pour transférer le message au fournisseur d'identité à travers l'agent d'utilisateur.

4) Le fournisseur d'identité identifie le principal

A l'étape 4, le principal est identifié par le fournisseur d'identité par un moyen en dehors du domaine d'application de ce profil. Cela peut exiger un nouvel acte d'authentification, ou il peut réutiliser une session authentifiée existante.

5) Le fournisseur d'identité produit une <Response> au fournisseur de service

A l'étape 5, le fournisseur d'identité produit un message `<Response>` à délivrer par l'agent d'utilisateur au fournisseur de service. La liaison HTTP POST, ou HTTP Artifact peut être utilisée pour transférer le message au fournisseur de service à travers l'agent d'utilisateur. Le message peut indiquer une erreur, ou incluera (au moins) une assertion d'authentification. La liaison HTTP Redirect ne doit pas être utilisée, car la réponse excèdera normalement la longueur d'URL permise par la plupart des agents d'utilisateur.

6) Le fournisseur de service accorde ou refuse l'accès au principal

A l'étape 6, ayant reçu la réponse du fournisseur d'identité, le fournisseur de service peut répondre à l'agent d'utilisateur du principal par sa propre erreur, ou il peut établir son propre contexte de sécurité pour le principal et retourner la ressource demandée.

Un fournisseur d'identité peut initier ce profil à l'étape 5 et produire un message `<Response>` à un fournisseur de service sans les étapes précédentes.

11.4.1.3 Description de profil

Si le profil est initialisé par le fournisseur de service, commencer par le § 11.4.1.3.1. S'il est initialisé par le fournisseur d'identité, commencer par le § 11.4.1.3.5. Dans les descriptions ci-dessous, ce qui suit se réfère à:

Service de signature unique

C'est le point d'extrémité de protocole de demande d'authentification au fournisseur d'identité auquel le message <AuthnRequest> (ou l'artifice qui le représente) est délivré par l'agent d'utilisateur.

Service de consommateur d'assertion

C'est le point d'extrémité de protocole de demande d'authentification au fournisseur de service auquel le message <Response> (ou l'artifice qui le représente) est délivré par l'agent d'utilisateur.

11.4.1.3.1 Demande HTTP au fournisseur de service

Si le premier accès est au fournisseur de service, une demande arbitraire de ressource peut initialiser le profil. Il n'y a pas de restriction sur la forme de la demande. Le fournisseur de service est libre d'utiliser tout moyen à sa convenance pour associer les interactions ultérieures à la demande d'origine. Chacune des liaisons fournit un mécanisme RelayState que le fournisseur de service peut utiliser pour associer l'échange de profils avec la demande d'origine. Le fournisseur de service devrait révéler aussi peu que possible de la demande dans la valeur RelayState sauf si l'utilisation du profil n'exige pas de telles mesures de confidentialité.

11.4.1.3.2 Le fournisseur de service détermine le fournisseur d'identité

Cette étape dépend de l'implémentation. Le fournisseur de service peut utiliser le profil de découverte de fournisseur d'identité SAML, décrit au § 11.4.3. Le fournisseur de service peut aussi choisir de rediriger l'agent d'utilisateur sur un autre service qui soit capable de déterminer un fournisseur d'identité approprié. Dans un tel cas, le fournisseur de service peut produire un `<AuthnRequest>` (comme dans l'étape suivante) à ce service pour être relayé au fournisseur d'identité, ou il peut s'appuyer sur le service intermédiaire pour produire un message `<AuthnRequest>` en son nom.

11.4.1.3.3 <AuthnRequest> est produit par le fournisseur de service au fournisseur d'identité

Une fois qu'un fournisseur d'identité est choisi, la localisation de son service de signature unique est déterminé, sur la base de la liaison SAML choisie par le fournisseur de service, pour envoyer le `<AuthnRequest>`. Des métadonnées peuvent être utilisées à cette fin. En réponse à une demande HTTP par l'agent d'utilisateur, une réponse HTTP est retournée qui contient un message `<AuthnRequest>` ou un artifice, selon la liaison SAML utilisée, pour être délivrée au service de signature unique du fournisseur d'identité.

Le format exact de cette réponse HTTP et de la demande HTTP suivante au service de signature unique est défini par la liaison SAML utilisée. Les règles spécifiques de profil pour le contenu du message `<AuthnRequest>` sont incluses dans le § 11.4.1.4.1. Si la liaison HTTP Redirect ou POST est utilisée, le message `<AuthnRequest>` est délivré directement au fournisseur d'identité dans cette étape. Si la liaison HTTP Artifact est utilisée, le profil de résolution d'artifice défini au § 11.4.6 est utilisé par le fournisseur d'identité, qui fait un rappel au fournisseur de service pour restituer le message `<AuthnRequest>`, en utilisant, par exemple, la liaison SOAP.

Il est recommandé que les échanges HTTP de cette étape soient faits sur TLS 1.0 pour conserver la confidentialité et l'intégrité du message. Le message `<AuthnRequest>` peut être signé, si l'authentification du producteur de la demande est requise. La liaison HTTP Artifact, si elle est utilisée, fournit aussi un moyen de remplacement d'authentification du producteur de la demande lorsque l'artifice est déréférencé.

Le fournisseur d'identité doit traiter le message `<AuthnRequest>` comme décrit dans la présente Recommandation. Cela peut exercer une contrainte sur les interactions ultérieures avec l'agent d'utilisateur, par exemple si l'attribut `IsPassive` est inclus.

11.4.1.3.4 Le fournisseur d'identité identifie le principal

Durant l'étape précédente ou à sa suite, le fournisseur d'identité doit à chaque fois établir l'identité du principal (sauf s'il retourne une erreur au fournisseur de service). L'attribut `ForceAuthn` `<AuthnRequest>`, s'il est présent avec une valeur de vrai, oblige le fournisseur d'identité à établir de nouveau cette identité, plutôt que de s'appuyer sur une session

existante qu'il pourrait avoir avec le principal. Autrement, et de tous les autres points de vue, le fournisseur d'identité peut utiliser tout moyen pour authentifier l'agent d'utilisateur, sous réserve de toutes exigences incluses dans le `<AuthnRequest>` sous la forme de l'élément `<RequestedAuthnContext>`.

11.4.1.3.5 Le fournisseur d'identité produit <Response> au fournisseur de service

Sans considération du succès ou de l'échec de `<AuthnRequest>`, le fournisseur d'identité devrait produire à l'agent d'utilisateur une réponse HTTP contenant un message `<Response>` ou un artifice, selon la liaison SAML utilisée, à délivrer au service consommateur d'assertions du fournisseur de service.

Le format exact de cette réponse HTTP et de la demande HTTP ultérieure au service consommateur d'assertion est défini par la liaison SAML utilisée. Les règles spécifiques du profil sur le contenu de `<Response>` sont incluses au § 11.4.1.4.2. Si la liaison HTTP POST est utilisée, le message `<Response>` est délivré directement au fournisseur de service dans cette étape. Si la liaison HTTP Artifact est utilisée, le profil de résolution d'artifice défini au § 11.4.6 est utilisé par le fournisseur de service, qui fait un rappel au fournisseur d'identité pour restituer le message `<Response>`, en utilisant par exemple la liaison SOAP.

La localisation du service consommateur d'assertions peut être déterminée en utilisant des métadonnées. Le fournisseur d'identité doit avoir des moyens d'établir que cette localisation est en fait contrôlée par le fournisseur de service. Un fournisseur de service peut indiquer la liaison SAML et le service consommateur d'assertions spécifique à utiliser dans sa `<AuthnRequest>` et le fournisseur d'identité doit les honorer s'il peut.

Il est recommandé que les demandes HTTP à cette étape soient faites sur TLS 1.0 pour conserver la confidentialité et l'intégrité du message. Le ou les éléments `<Assertion>` dans la `<Response>` doivent être signés si la liaison HTTP POST est utilisée, et peuvent être signés si la liaison HTTP-Artifact est utilisée.

Le fournisseur de service doit traiter le message `<Response>` et tout élément <Assertion> enchassé, comme décrit dans la présente Recommandation.

11.4.1.3.6 Le fournisseur de service accorde ou refuse l'accès à l'agent d'utilisateur

Pour compléter le profil, le fournisseur de service traite le ou les `<Response>` et `<Assertion>` et accorde ou refuse l'accès à la ressource. Le fournisseur de service peut établir un contexte de sécurité avec l'agent d'utilisateur en utilisant tout mécanisme de session qu'il choisit. Toute utilisation ultérieure de la ou des `<Assertion>` fournies est à la discrétion du fournisseur de service et des autres consommateurs d'assertions, sous réserve de toute restriction d'utilisation qu'elles contiennent.

11.4.1.4 Utilisation du protocole de demande d'authentification

Ce profil se fonde sur le protocole de demande d'authentification défini dans la présente Recommandation. Ici, le fournisseur de service est le producteur de la demande et le consommateur d'assertions, et le principal est le présentateur, le sujet demandé, et l'entité de confirmation. Il peut y avoir des consommateurs d'assertions ou des entités de confirmation supplémentaires, à la discrétion du fournisseur d'identité.

11.4.1.4.1 Utilisation de <AuthnRequest>

Un fournisseur de service peut inclure tout contenu de message comme décrit dans la présente Recommandation. Toutes les règles de traitement sont celles définies dans la présente Recommandation. L'élément `<Issuer>` doit être présent et doit contenir l'identifiant unique du fournisseur de service demandeur; l'attribut `Format` doit être omis ou avoir une valeur de `urn:oasis:names:tc:SAML:2.0:nameid-format:entity`.

Si le fournisseur d'identité ne peut ou ne veut pas satisfaire la demande, il doit répondre par un message `<Response>` contenant un ou des codes d'état d'erreur appropriés.

Si le fournisseur de service souhaite permettre au fournisseur d'identité d'établir un nouvel identifiant pour le principal, s'il n'en existe pas, il doit inclure un élément `<NameIDPolicy>` avec l'attribut `AllowCreate` réglé à "vrai". Autrement, seul un principal pour lequel le fournisseur d'identité avait précédemment établi un identifiant utilisable par le fournisseur de service peut être authentifié avec succès.

Le fournisseur de service peut inclure un élément `<Subject>` dans la demande qui nomme l'identité réelle sur laquelle il souhaite recevoir une assertion. Cet élément ne doit pas contenir d'élément `<SubjectConfirmation>`. Si le fournisseur d'identité ne reconnaît pas le principal comme ayant cette identité, il doit alors répondre par un message `<Response>` contenant un état d'erreur et pas d'assertion.

Le message `<AuthnRequest>` peut être signé (selon les indications de la liaison SAML utilisée). Si la liaison HTTP Artifact est utilisée, l'authentification des parties est facultative et tout mécanisme permis par la liaison peut être utilisé.

Si la <AuthnRequest> n'est pas authentifiée et/ou protégée en intégrité, on ne doit pas se fier aux informations qu'elle contient sauf à titre de conseil. Que la demande soit signée ou non, le fournisseur d'identité doit s'assurer que tout élément <AssertionConsumerServiceURL> ou <AssertionConsumerServiceIndex> dans la demande est vérifié comme appartenant au fournisseur de service auquel la réponse sera envoyée. Ne pas le faire peut avoir pour résultat une attaque par intrusion.

11.4.1.4.2 Utilisation de <Response>

NOTE 1 (informative) – PE26 (voir OASIS PE:2006) propose des éclaircissements pour le présent paragraphe, voir les précisions à l'Appendice VIII.

Si le fournisseur d'identité souhaite retourner une erreur, il ne doit pas inclure d'assertion dans le message <Response>. Autrement, si la demande est réussie (ou si la réponse n'est pas associée à une demande), l'élément <Response> doit se conformer à ce qui suit:

- L'élément <Issuer> peut être omis, mais s'il est présent, il doit contenir l'identifiant unique du fournisseur d'identité producteur; l'attribut Format doit être omis ou avoir une valeur de urn:oasis:names:tc:SAML:2.0:nameid-format:entity.

  NOTE 2 (informative) – PE17 (voir OASIS PE:2006) suggère de remplacer le paragraphe ci-dessus par:

  Si le message <Response> est signé ou si une assertion enchassée est chiffrée, l'élément <Issuer> doit alors être présent. Autrement il peut être omis. S'il est présent, il doit contenir l'identifiant unique du fournisseur d'identité producteur; l'attribut Format doit être omis ou avoir une valeur de urn:oasis:names:tc:SAML:2.0:nameid-format:entity.
- Il doit contenir au moins une <Assertion>. Chaque élément <Issuer> d'assertion doit contenir l'identifiant unique du fournisseur d'identité producteur; l'attribut Format doit être omis ou avoir une valeur de urn:oasis:names:tc:SAML:2.0:nameid-format:entity.
- L'ensemble d'une ou plusieurs assertions doit contenir au moins une <AuthnStatement> qui reflète l'authentification du principal auprès du fournisseur d'identité.
- Au moins une assertion contenant une <AuthnStatement> doit contenir un élément <Subject> avec au moins un élément <SubjectConfirmation> contenant un Method de urn:oasis:names:tc:SAML:2.0:cm:bearer. Si le fournisseur d'identité accepte le profil Single Logout, défini au § 11.4.4, une telle déclaration d'authentification doit inclure un attribut SessionIndex pour activer la demande de terminaison de session pour la session par le fournisseur de service.
- L'élément <SubjectConfirmation> du titulaire décrit ci-dessus doit contenir un élément <SubjectConfirmationData> qui contient un attribut Recipient contenant l'URL de service de consommateur d'assertion du fournisseur de service et un attribut NotOnOrAfter qui limite la fenêtre durant laquelle l'assertion peut être délivrée. Il peut contenir un attribut Address limitant l'adresse de client à partir de laquelle l'assertion peut être délivrée. Il ne doit pas contenir d'attribut NotBefore. Si le message contenant est en réponse à une <AuthnRequest>, l'attribut InResponseTo doit alors correspondre à ID (l'identifiant) de la demande.
- D'autres déclarations et méthodes de confirmation peuvent être incluses dans la ou les assertions, à la discrétion du fournisseur d'identité. En particulier, les éléments <AttributeStatement> peuvent être inclus. Le <AuthnRequest> peut contenir un attribut XML AttributeConsumingServiceIndex référençant les informations sur les attributs désirés ou requis comme au § 9. Le fournisseur d'identité peut l'ignorer, ou envoyer d'autres attributs à sa discrétion.
- La ou les assertions contenant une confirmation du sujet du titulaire doivent contenir une <AudienceRestriction> incluant l'identifiant unique du fournisseur de service comme un élément <Audience>.
- D'autres conditions (et d'autres éléments <Audience>) peuvent être inclus à la demande du fournisseur de service ou à la discrétion du fournisseur d'identité. (Bien sûr, de telles conditions doivent toutes être comprises et acceptées par le fournisseur de service afin que l'assertion soit considérée comme valide.) Le fournisseur d'identité n'est pas obligé d'honorer l'ensemble de <Conditions> demandées dans la <AuthnRequest>, s'il en est.

11.4.1.4.3 Règles de traitement du message <Response>

NOTE (informative) – PE26 (voir OASIS PE:2006) propose des éclaircissements pour le présent paragraphe, voir les précisions à l'Appendice VIII.

Quelles que soit la liaison SAML utilisée, le fournisseur de service doit faire ce qui suit:

- vérifier toutes les signatures présentes sur la ou les assertions ou réponses;

- vérifier que l'attribut `Recipient` de tout `<SubjectConfirmationData>` de titulaire correspond à l'URL de consommateur d'assertion auquel le `<Response>` ou l'artifice a été délivré;
- vérifier que l'attribut `NotOnOrAfter` dans tout `<SubjectConfirmationData>` du titulaire n'est pas dépassé, sous réserve du biais d'horloge admissible entre les fournisseurs;
- vérifier que l'attribut `InResponseTo` dans le `<SubjectConfirmationData>` du titulaire est égal à l'identifiant de son message `<AuthnRequest>` d'origine, à moins que la réponse ne soit pas sollicitée, auquel cas l'attribut ne doit pas être présent;
- vérifier que toute assertion sur laquelle on s'appuie est valide à tous autres égards;
- si un `<SubjectConfirmationData>` du titulaire inclut un attribut `Address`, le fournisseur de service peut la confronter à l'adresse de client de l'agent d'utilisateur;
- toute assertion non valide, ou dont les exigences de confirmation de sujet ne peuvent pas être satisfaites devrait être éliminée et ne devrait pas être utilisée pour établir un contexte de sécurité pour le principal;
- si un `<AuthnStatement>` utilisé pour établir un contexte de sécurité pour le principal contient un attribut `SessionNotOnOrAfter`, le contexte de sécurité devrait être éliminé une fois ce moment arrivé, à moins que le fournisseur de service ne rétablisse l'identité du principal en répétant l'utilisation de ce profil.

11.4.1.4.4 Règles de traitement de message <Response> spécifique de l'artifice

Si la liaison HTTP Artifact est utilisée pour délivrer la `<Response>`, le déréférencement de l'artifice en utilisant le profil de résolution d'artifice doit être mutuellement authentifié, protégé en intégrité, et confidentiel.

Le fournisseur d'identité doit s'assurer que seul le fournisseur de service auquel le message `<Response>` a été fourni reçoit le message en résultat d'une demande `<ArtifactResolve>`.

La liaison SAML utilisée pour déréférencer l'artifice ou les signatures de messages peuvent être utilisées pour authentifier les parties et protéger le messages.

11.4.1.4.5 Règles de traitement spécifiques de POST

NOTE (informative) – PE26 (voir OASIS PE:2006) propose des éclaircissements pour le présent paragraphe, voir les précisions à l'Appendice VIII.

Si la liaison HTTP POST est utilisée pour délivrer la `<Response>`, la ou les assertions incluses doivent être signées.

Le fournisseur de service doit s'assurer que les assertions de titulaire ne sont pas répétées, en conservant l'ensemble des valeurs `ID` (d'identifiant) utilisées pendant la durée où l'assertion serait considérée comme valide sur la base de l'attribut `NotOnOrAfter` dans le `<SubjectConfirmationData>`.

11.4.1.5 Réponses non sollicitées

Un fournisseur d'identité peut initialiser ce profil en délivrant un message `<Response>` non sollicité à un fournisseur de service.

Une `<Response>` non sollicitée ne doit pas contenir d'attribut `InResponseTo`, et aucun élément `<SubjectConfirmationData>` de titulaire non plus. Si des métadonnées sont utilisées, la `<Response>` ou l'artifice devrait être délivrée au point d'extrémité `<md:AssertionConsumerService>` du fournisseur de service désigné par défaut.

On doit mentionner particulièrement que le fournisseur d'identité peut inclure un paramètre "RelayState" spécifique de la liaison qui indique, sur la base d'un accord mutuel avec le fournisseur de service, comment traiter les interactions ultérieures avec l'agent d'utilisateur. Cela peut être l'URL d'une ressource chez le fournisseur de service. Le fournisseur de service devrait être prêt à traiter des réponses non sollicitées en désignant une localisation par défaut où envoyer l'agent d'utilisateur suite au traitement réussi d'une réponse.

11.4.1.6 Utilisation de métadonnées

Le paragraphe 11.4.2.5 définit un élément de point d'extrémité, `<md:SingleSignOnService>`, pour décrire les liaisons et localisations acceptées auxquelles un fournisseur de service peut envoyer des demandes à un fournisseur d'identité utilisant ce profil.

L'attribut `WantAuthnRequestsSigned` de l'élément `<md:IDPSSODescriptor>` peut être utilisé par un fournisseur d'identité pour informer de l'exigence de signature de la demande. L'attribut `AuthnRequestsSigned` de l'élément `<md:SPSSODescriptor>` peut être utilisé par un fournisseur de service pour informer de l'intention de signer toutes ses demandes.

Les fournisseurs peuvent documenter la ou les clés utilisées pour signer les demandes, réponses, et assertions au moyen des éléments <md:KeyDescriptor> avec un attribut d'utilisation de sign. Lors du chifrement d'éléments SAML, les éléments <md:KeyDescriptor> avec un attribut d'utilisation de encrypt peuvent être utilisés pour documenter les algorithmes de chiffrement et les réglages acceptés, et les clés publiques utilisées pour recevoir les clés de chiffrement brutes.

L'élément de point d'extrémité indexé <md:AssertionConsumerService> est utilisé pour décrire les liaisons et localisations acceptées pour lesquelles un fournisseur d'identité peut envoyer des réponses à un fournisseur de service utilisant ce profil. L'attribut index est utilisé pour distinguer les points d'extrémité possibles qui peuvent être spécifiés par référence dans le message <AuthnRequest>. L'attribut isDefault est utilisé pour spécifier le point d'extrémité à utiliser si il n'est pas spécifié dans une demande.

L'attribut WantAssertionsSigned de l'élément <md:SPSSODescriptor> peut être utilisé par un fournisseur de service pour informer de l'exigence que les assertions délivrées avec ce profil soient signées. Ceci s'ajoute à toute exigence de signature imposée par l'utilisation d'une liaison particulière. Le fournisseur d'identité n'est pas obligé par cela, mais il est averti qu'il est vraisemblable qu'une assertion non signée ne sera pas suffisante.

Si la demande ou message de réponse est délivré en utilisant la liaison HTTP Artifact, le producteur de l'artifice doit fournir au moins un élément de point d'extrémité <md:ArtifactResolutionService> dans ses métadonnées.

Le <md:IDPSSODescriptor> peut contenir les éléments <md:NameIDFormat>, <md:AttributeProfile>, et <saml:Attribute> pour indiquer la capacité générale à prendre en charge des formats d'identifiant de nom, des profils d'attribut particuliers, ou des attributs et valeurs spécifiques. La capacité à prendre en charge l'une de ces caractéristiques durant un échange d'authentification donné dépend de la politique et est à la discrétion du fournisseur d'identité.

L'élément <md:SPSSODescriptor> peut aussi être utilisé pour informer du besoin ou du désir du fournisseur de service que les attributs SAML soient délivrés avec des informations d'authentification. L'inclusion réelle des attributs est toujours à la discrétion du fournisseur d'identité. Un ou plusieurs éléments <md:AttributeConsumingService> peuvent être inclus dans ses métadonnées, chacun ayant un attribut index pour distinguer différents services qui peuvent être spécifiés par référence dans le message <AuthnRequest>. L'attribut isDefault est utilisé pour spécifier un ensemble par défaut des exigences d'attribut.

11.4.2 Profil de client ou mandataire amélioré (ECP)

Un client ou mandataire amélioré (ECP, *enhanced client or proxy*) est une entité système qui sait comment contacter un fournisseur d'identité approprié, éventuellement d'une manière dépendante du contexte, et aussi prendre en charge la liaison SOAP inverse (PAOS) (voir le § 10).

Un exemple de scénario activé par ce profil est le suivant: un principal, maniant un ECP, l'utilise pour accéder à une ressource chez un fournisseur de service, ou accéder à un fournisseur d'identité tel que le fournisseur de service et la ressource désirée soient compris ou implicites. Le principal s'authentifie (ou s'est déjà authentifié) auprès du fournisseur d'identité, qui produit alors une assertion d'authentification (éventuellement avec des apports du fournisseur de service). Le fournisseur de service consomme alors l'assertion et établit ensuite un contexte de sécurité pour le principal. Durant ce processus, un identifiant de nom peut aussi être établi entre les fournisseurs pour le principal, sous réserve des paramètres de l'interaction et du consentement du principal.

Ce profil se fonde sur le protocole de demande d'authentification SAML en conjonction avec la liaison PAOS.

NOTE – Les moyens par lesquels un principal s'authentifie auprès d'un fournisseur d'identité sont en dehors du domaine d'application de SAML.

11.4.2.1 Informations requises

Identification: urn:oasis:names:tc:SAML:2.0:profiles:SSO:ecp (c'est aussi l'espace de nom cible alloué dans le schéma de profil ECP correspondant à l'Annexe A.

Informations de contact: security-services-comment@lists.oasis-open.org

Identifiants de méthode de confirmation SAML: l'identifiant de méthode de confirmation "bearer" de SAML V2.0, urn:oasis:names:tc:SAML:2.0:cm:bearer, est utilisé par ce profil.

Description: donnée ci-dessous

Mises à jour: aucune.

11.4.2.2 Aperçu général sur le profil

Comme présenté ci-dessus, le profil ECP spécifie les interactions entre clients ou mandataires améliorés et fournisseurs de services et fournisseurs d'identités. C'est une application spécifique du profil SSO décrit au § 11.4.1. Si elles ne sont

pas autrement spécifiées par ce profil, et si elles ne sont pas spécifiques de l'utilisation de liaisons fondées sur un navigateur, les règles spécifiées au § 11.4.1 doivent être observées.

Un ECP est un client ou mandataire qui satisfait aux deux conditions suivantes:

- il a, ou sait comment obtenir, des informations sur le fournisseur d'identité que le principal associé à l'ECP souhaite utiliser, dans le contexte d'une interaction avec un fournisseur de service.

Cela permet à un fournisseur de service de faire une demande d'authentification à l'ECP sans qu'il ait besoin de savoir ou de découvrir le fournisseur d'identité approprié (en sautant effectivement l'étape 2 du profil SSO du § 11.4.1).

- Il est capable d'utiliser une liaison SOAP inversée (PAOS) comme profilée ici pour une demande et réponse d'authentification.

Cela permet à un fournisseur de service d'obtenir une assertion d'authentification via un ECP qui n'est pas autrement (c'est-à-dire en dehors du contexte de l'interaction imméiate) nécessairement directement adressable ni continûment disponible. Cela démultiplie aussi les bénéfices de SOAP tout en utilisant un schéma et profil d'échange bien défini pour améliorer l'interopérabilité. L'ECP peut être vu comme un intermédiaire SOAP entre le fournisseur de service et le fournisseur d'identité.

Un *client amélioré* peut être un navigateur ou quelque autre agent d'utilisateur qui prend en charge la fonctionnalité décrite dans ce profil. Un *mandataire amélioré* est un mandataire HTTP qui émule un client amélioré. Sauf mention contraire, toutes les déclarations faisant référence à des clients améliorés sont à comprendre comme des déclarations à la fois sur les clients améliorés et sur les mandataires de client amélioré.

Comme le client amélioré envoie et reçoit des messages dans le corps des demandes et réponses HTTP, il n'y a pas de restriction arbitraire sur la taille des messages de protocole.

Ce profil démultiplie la liaison SOAP inversé (PAOS) (voir au § 10). Les implémentations de ce profil doivent suivre les règles sur les indications HTTP de prise en charge de PAOS spécifiées dans cette liaison, en plus de celles spécifiées dans ce profil. Ce profil utilise un bloc d'en-tête SOAP PAOS porté entre le répondeur HTTP et l'ECP mais ne définit pas PAOS lui-même. Ce profil définit les blocs d'en-tête SOAP qui accompagnent les demandes et réponses SAML. Ces blocs d'en-tête peuvent être composés avec autant d'autres blocs d'en-tête SOAP que nécessaire, par exemple, avec le bloc d'en-tête de sécurité de message SOAP pour ajouter des caractéristiques de sécurité si nécessaire, par exemple, une signature numérique ajoutée à la demande d'authentification.

Deux ensembles de blocs d'en-tête de demande/réponse SOAP sont utilisés: les blocs d'en-tête PAOS pour les informations PAOS génériques et les blocs d'en-tête spécifiques de profil ECP pour convoyer les informations spécifiques de fonctionnalité de profil ECP.

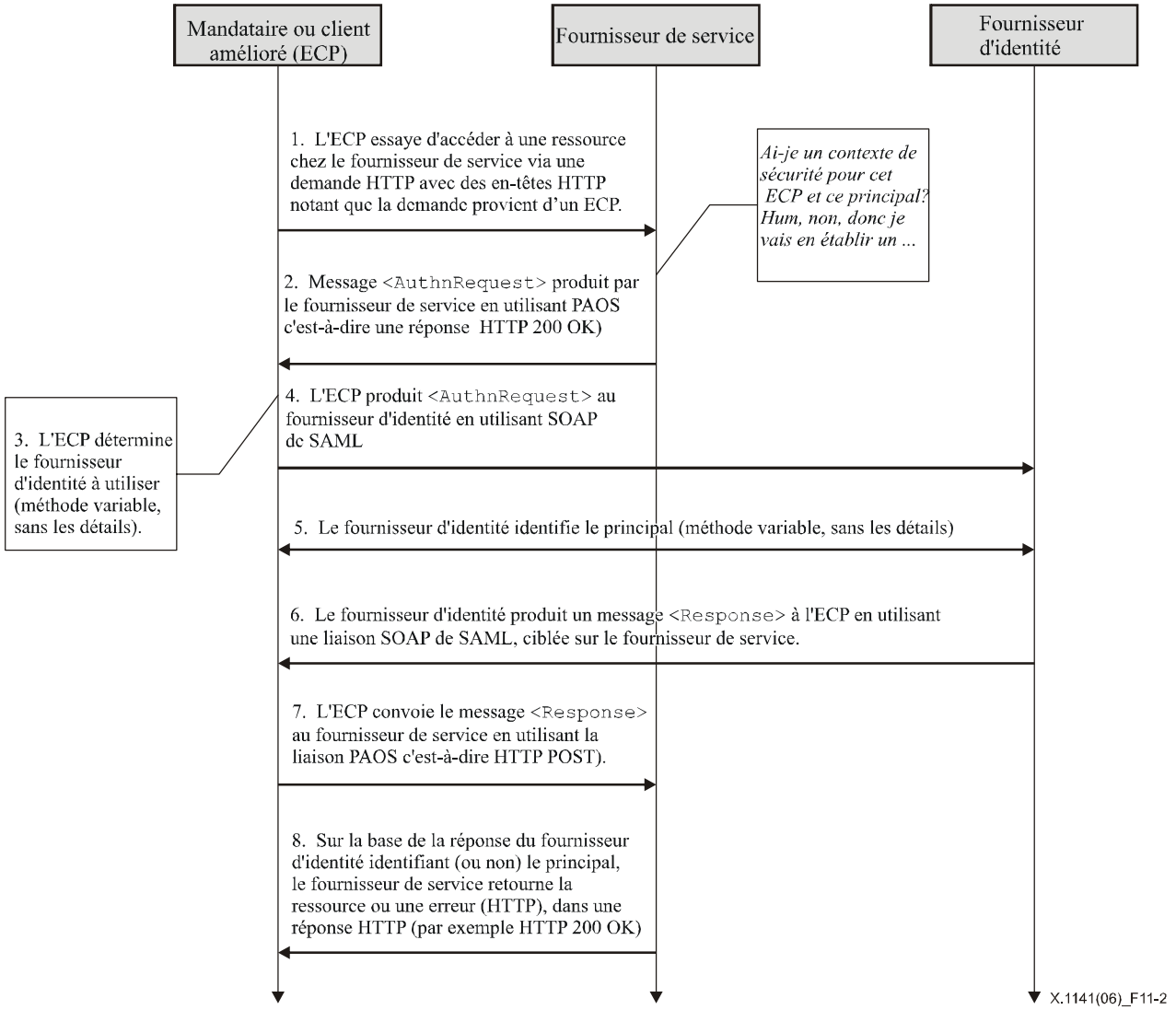

Figure 11-2/X.1141 – Flux de traitement dans le profil ECP

La Figure 11-2 illustre l'organigramme de base pour SSO utilisant un ECP. Les étapes suivantes sont décrites par le profil. Dans une étape individuelle, il peut y avoir un ou plusieurs échanges de messages réels selon la liaison utilisée pour cette étape et d'autres selon le comportement de l'implémentation.

1) ECP produit une demande HTTP au fournisseur de service

A l'étape 1, le principal, via un ECP, fait une demande HTTP de ressource sécurisée à un fournisseur de service, où le fournisseur de service n'a pas de contexte de sécurité établi pour l'ECP et le principal.

2) Le fournisseur de service produit <AuthnRequest> à l'ECP

A l'étape 2, le fournisseur de service produit un message `<AuthnRequest>` à l'ECP, qui est à délivrer par l'ECP au fournisseur d'identité approprié. On utilise ici la liaison SOAP inverse (PAOS) (voir le § 10).

3) ECP détermine le fournisseur d'identité

A l'étape 3, l'ECP obtient la localisation d'un point d'extrémité chez un fournisseur d'identité pour le protocole de demande d'authentification qui prend en charge sa liaison préférée. Les moyens par lesquels ceci est réalisé dépendent de l'implémentation. L'ECP peut utiliser le profil de découverte de fournisseur d'identité SAML décrit au § 11.4.3.

NOTE (informative) – PE18 (voir OASIS PE:2006) suggère de supprimer la dernière phrase de l'alinéa ci-dessus.

4) ECP convoie <AuthnRequest> au fournisseur d'identité

A l'étape 4, l'ECP convoie le `<AuthnRequest>` au fournisseur d'identité identifié à l'étape 3 en utilisant une forme modifiée de la liaison SOAP de SAML (voir le § 10) avec la tolérance supplémentaire que le fournisseur d'identité peut échanger des messages HTTP arbitraires avec l'ECP avant de répondre à la demande SAML.

5) Le fournisseur d'identité identifie le principal

A l'étape 5, le principal est identifié par le fournisseur d'identité par des moyens qui sortent du domaine d'application de ce profil. Cela peut exiger un nouvel acte d'authentification, ou il peut réutiliser une session authentifiée exixtante.

6) Le fournisseur d'identité produit <Response> à l'ECP, visant le fournisseur de service

A l'étape 6, le fournisseur d'identité produit un message `<Response>`, en utilisant la liaison SOAP de SAML, à livrer par l'ECP au fournisseur de service. Le message peut indiquer une erreur, ou incluera (au moins) une assertion d'authentification.

7) ECP convoie un message <Response> au fournisseur de service

A l'étape 7, l'ECP convoie le message `<Response>` au fournisseur de service en utilisant la liaison PAOS.

8) Le fournisseur de service accorde ou refuse l'accès au principal

A l'étape 8, ayant reçu le message `<Response>` de la part du fournisseur d'identité, le fournisseur de service établit son propre contexte de sécurité pour le principal et retourne la ressource demandée, ou répond à l'ECP du principal par une erreur.

11.4.2.3 Description de profil

Les paragraphes qui suivent donnent des définitions détaillées des étapes individuelles.

11.4.2.3.1 L'ECP produit une demande HTTP au fournisseur de service

L'ECP envoie une demande HTTP à un fournisseur de service, spécifiant une ressource. Cette demande HTTP doit être conforme à la liaison PAOS, ce qui signifie qu'elle doit inclure les champs d'en-tête HTTP suivants:

1) le champ d'en-tête HTTP `Accept` qui indique la capacité à accepter le type MIME "`application/vnd.paos+xml`".
2) le champ d'en-tête HTTP `PAOS` qui spécifie la version `PAOS` avec `urn:liberty:paos:2003-08` au minimum.
3) de plus, la prise en charge de ce profil doit être spécifiée dans le champ d'en-tête HTTP `PAOS` comme une valeur de service, avec la valeur `urn:oasis:names:tc:SAML:2.0:profiles:SSO:ecp`. Cette valeur devrait correspondre à l'attribut de service dans le bloc d'en-tête SOAP de demande PAOS.

Par exemple, un agent d'utilisateur peut demander une page d'un fournisseur de service comme suit:

```
GET /index HTTP/1.1
Host: identity-service.example.com
Accept: text/html; application/vnd.paos+xml
PAOS: ver='urn:liberty:paos:2003-08' ;
'urn:oasis:names:tc:SAML:2.0:profiles:SSO:ecp'
```

11.4.2.3.2 Le fournisseur de service produit une <AuthnRequest> à l'ECP

Lorsque le fournisseur de service exige un contexte de sécurité pour le principal avant de permettre l'accès à la ressource spécifiée, c'est-à-dire, avant de fournir un service ou des données, il peut répondre à la demande HTTP en utilisant la liaison PAOS avec un message `<AuthnRequest>` dans la réponse HTTP. Le fournisseur de service produira une réponse HTTP 200 OK à l'ECP qui contient une seule enveloppe SOAP.

L'enveloppe SOAP doit contenir:

1) un élément `<AuthnRequest>` dans le corps SOAP, destiné au dernier receveur SOAP, le fournisseur d'identité;
2) un bloc d'en-tête SOAP PAOS visant l'ECP en utilisant la valeur d'acteur SOAP de `http://schemas.xmlsoap.org/soap/actor/next`. Ce bloc d'en-tête fournit des informations de contrôle telles que l'URL auquel envoyer la réponse dans ce schéma d'échange de messages à réponse sollicitée;

3) un bloc d'en-tête SOAP de demande spécifique de profil ECP visant l'ECP en utilisant l'acteur SOAP `http://schemas.xmlsoap.org/soap/actor/next`. Le bloc d'en-tête de demande ECP définit les informations qui se rapportent à la demande d'authentification que l'ECP peut avoir besoin de traiter, comme une liste des fournisseurs d'identité acceptables pour le fournisseur de service, si l'ECP peut interagir avec le principal à travers le client, et le nom, lisible par l'homme, du fournisseur de service qui peut être affiché au principal.

L'enveloppe SOAP peut contenir un bloc d'en-tête SOAP RelayState d'ECP visant l'ECP en utilisant la valeur d'acteur SOAP de `http://schemas.xmlsoap.org/soap/actor/next`. L'en-tête contient des informations d'état à retourner par l'ECP avec la réponse SOAP.

11.4.2.3.3 L'ECP détermine le fournisseur d'identité

L'ECP va déterminer quel fournisseur d'identité est approprié et acheminer le message SOAP en conséquence.

11.4.2.3.4 L'ECP produit une <AuthnRequest> au fournisseur d'identité

L'ECP doit retirer les blocs d'en-tête PAOS, ECP RelayState, et ECP Request, avant de passer le message `<AuthnRequest>` au fournisseur d'identité, en utilisant une forme modifiée de la liaison SOAP de SAML. La demande SAML est soumise via SOAP de la façon habituelle, mais le fournisseur d'identité peut répondre à la demande HTTP de l'ECP par une réponse HTTP contenant, par exemple, une forme de connexion HTML ou autre réponse orientée présentation. Une séquence d'échanges HTTP peut avoir lieu, mais finalement, le fournisseur d'identité doit achever l'échange SOAP de SAML et retourner une réponse SAML via la liaison SOAP.

L'élément `<AuthnRequest>` peut lui-même être signé par le fournisseur de service. A cet égard, les règles de message spécifiées dans le profil SSO de navigateur au paragraphe 11.4.1.4.1 doivent être suivies.

Avant cette étape, ou à sa suite, le fournisseur d'identité doit établir d'une façon quelconque l'identité du principal, ou il doit retourner une erreur `<Response>`, comme décrit au paragraphe 11.4.2.3.6 ci-dessous.

11.4.2.3.5 Le fournisseur d'identité identifie le principal

A tout moment pendant l'étape précédente ou à sa suite, le fournisseur d'identité doit établir l'identité du principal (sauf s'il retourne une erreur au fournisseur de service). L'attribut `ForceAuthn <AuthnRequest>`, s'il est présent avec une valeur de vrai, oblige le fournisseur d'identité à établir à nouveau cette identité, plutôt que de s'appuyer sur une session existante qu'il pourrait avoir avec le principal. Autrement, et sous tous les autres points de vue, le fournisseur d'identité peut utiliser tout moyen pour authentifier l'agent d'utilisateur, sous réserve de toutes les exigences incluses dans le `<AuthnRequest>` sous la forme de l'élément `<RequestedAuthnContext>`.

11.4.2.3.6 Le fournisseur d'identité produit une <Response> à l'ECP, visant le fournisseur de service

Le fournisseur d'identité retourne un message SAML `<Response>` (ou une faute SOAP) lorsqu'il est présenté avec une demande d'authentification, après avoir établi l'identité du principal. La réponse SAML est convoyée en utilisant la liaison SOAP SAML dans un message SOAP avec un élément `<Response>` dans le corps SOAP, destiné au fournisseur de service comme receveur SOAP ultime. Les règles pour la réponse spécifiées dans le profil de navigateur SSO au § 11.4.1.4.2 doivent être suivies.

Le message de réponse du fournisseur d'identité doit contenir un bloc d'en-tête SOAP de réponse d'ECP spécifique du profil, et peut contenir un bloc d'en-tête RelayState d'ECP, tous deux visant l'ECP.

11.4.2.3.7 L'ECP convoie le message <Response> au fournisseur de service

L'ECP retire le ou les blocs d'en-tête, et peut ajouter un bloc d'en-tête SOAP de réponse PAOS et un bloc d'en-tête RelayState d'ECP avant de transmettre la réponse SOAP au fournisseur de service en utilisant la liaison PAOS.

Le bloc d'en-tête SOAP `<paos:Response>` dans la réponse au fournisseur de service est généralement utilisée pour corréler cette réponse à une demande antérieure du fournisseur de service. Dans ce profil, l'attribut `refToMessageID` de corrélation n'est pas nécessaire car l'attribut `InResponseTo` de l'élément `<Response>` de SAML peut être utilisé à cette fin, mais si le bloc d'en-tête SOAP `<paos:Request>` avait un `messageID`, le bloc d'en-tête SOAP `<paos:Response>` doit alors être utilisé.

La valeur de bloc d'en-tête `<ecp:RelayState>` est normalement fournie par le fournisseur de service à l'ECP avec sa demande, mais si le fournisseur d'identité est en train de produire une réponse non sollicitée (sans avoir reçu une demande SAML correspondante), il peut alors inclure un bloc d'en-tête RelayState qui indique, sur la base d'un accord mutuel avec le fournisseur de service, comment traiter les interactions suivantes avec l'ECP. Cela peut être l'URL d'une ressource chez le fournisseur de service.

Si le fournisseur de service avait inclus un bloc d'en-tête SOAP `<ecp:RelayState>` dans sa demande à l'ECP, ou si le fournisseur d'identité avait inclus un bloc d'en-tête SOAP `<ecp:RelayState>` avec sa réponse, l'ECP doit alors inclure un bloc d'en-tête identique avec la réponse SOAP envoyée au fournisseur de service. La valeur du fournisseur de service pour ce bloc d'en-tête (s'il en est) doit avoir la priorité.

11.4.2.3.8 Le fournisseur de service accorde ou refuse l'accès au principal

Une fois que le fournisseur de service a reçu la réponse SOAP dans une demande HTTP (dans une enveloppe SOAP utilisant PAOS), il peut répondre par les données de service dans la réponse HTTP. En consommant la réponse, les règles spécifiées dans le profil SSO de navigateur aux paragraphes 11.4.1.4.3 et 11.4.1.4.5 doivent être suivies. C'est-à-dire que les mêmes règles de traitement utilisées lors de la réception de `<Response>` avec la liaison HTTP POST s'appliquent à l'utilisation de PAOS.

11.4.2.4 Utilisation du schéma de profil d'ECP

Le schéma XML de profil d'ECP définit les blocs d'en-tête demande/réponse SOAP utilisés par ce profil. Ci-après figure une liste complète de ce document de schéma.

```
<schema
    targetNamespace="urn:oasis:names:tc:SAML:2.0:profiles:SSO:ecp"
    xmlns="http://www.w3.org/2001/XMLSchema"
    xmlns:ecp="urn:oasis:names:tc:SAML:2.0:profiles:SSO:ecp"
    xmlns:samlp="urn:oasis:names:tc:SAML:2.0:protocol"
    xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion"
    xmlns:S="http://schemas.xmlsoap.org/soap/envelope/"
    elementFormDefault="unqualified"
    attributeFormDefault="unqualified"
    blockDefault="substitution"
    version="2.0">
    <import namespace="urn:oasis:names:tc:SAML:2.0:protocol"
        schemaLocation="saml-schema-protocol-2.0.xsd"/>
    <import namespace="urn:oasis:names:tc:SAML:2.0:assertion"
        schemaLocation="saml-schema-assertion-2.0.xsd"/>
    <import namespace="http://schemas.xmlsoap.org/soap/envelope/"
        schemaLocation="http://schemas.xmlsoap.org/soap/envelope/"/>
    <annotation>
        <documentation>
            Document identifier: saml-schema-ecp-2.0
            Location: http://docs.oasis-open.org/security/saml/v2.0/
            Revision history:
              V2.0 (March, 2005):
                Custom schema for ECP profile, first published in SAML 2.0.
        </documentation>
    </annotation>

    <element name="Request" type="ecp:RequestType"/>
    <complexType name="RequestType">
        <sequence>
            <element ref="saml:Issuer"/>
            <element ref="samlp:IDPList" minOccurs="0"/>
        </sequence>
        <attribute ref="S:mustUnderstand" use="required"/>
        <attribute ref="S:actor" use="required"/>
        <attribute name="ProviderName" type="string" use="optional"/>
        <attribute name="IsPassive" type="boolean" use="optional"/>
    </complexType>

    <element name="Response" type="ecp:ResponseType"/>
    <complexType name="ResponseType">
        <attribute ref="S:mustUnderstand" use="required"/>
        <attribute ref="S:actor" use="required"/>
        <attribute name="AssertionConsumerServiceURL" type="anyURI"
use="required"/>
    </complexType>

    <element name="RelayState" type="ecp:RelayStateType"/>
    <complexType name="RelayStateType">
        <simpleContent>
            <extension base="string">
                <attribute ref="S:mustUnderstand" use="required"/>
```

```
                <attribute ref="S:actor" use="required"/>
            </extension>
        </simpleContent>
    </complexType>
</schema>
```

Les paragraphes suivants décrivent comment ces constructions XML doivent être utilisées.

11.4.2.4.1 Bloc d'en-tête de demande PAOS: de SP à ECP

Le bloc d'en-tête de demande PAOS signale l'utilisation du traitement PAOS et inclut les attributs suivants:

– `responseConsumerURL` [Exigé]

  Spécifie où l'ECP doit envoyer une réponse d'erreur. Aussi utilisé pour vérifier la correction de la réponse du fournisseur d'identité, en croisant cette localisation avec le `AssertionServiceConsumerURL` dans le bloc d'en-tête de réponse de l'ECP. Cette valeur doit être la même que le `AssertionServiceConsumerURL` (ou l'URL référencé dans les métadonnées) convoyé dans le `<AuthnRequest>`.

  NOTE (informative) – PE22 (voir OASIS PE:2006) suggère de changer dans la dernière phrase `AssertionServiceConsumerURL en AssertionConsumerServiceURL`.

– `service` [Exigé]

  Indique que le service PAOS en cours d'utilisation est ce profil d'authentification SAML. La valeur doit être `urn:oasis:names:tc:SAML:2.0:profiles:SSO:ecp`.

– `SOAP-ENV:mustUnderstand` [Exigé]

  La valeur doit être 1 (vrai). Une faute SOAP doit être générée si le bloc d'en-tête PAOS n'est pas compris.

– `SOAP-ENV:actor` [Exigé]

  La valeur doit être `http://schemas.xmlsoap.org/soap/actor/next`.

– `messageID` [Facultatif]

  Permet la corrélation de réponse facultative. Il peut être utilisé dans ce profil, mais n'est pas exigé, car cette fonctionnalité est fournie par la couche de protocole SAML, via l'attribut ID dans l'attribut `<AuthnRequest>` et l'attribut `InResponseTo` de `<Response>`.

Le bloc d'en-tête SOAP de demande PAOS n'a pas d'élément contenu.

11.4.2.4.2 Bloc d'en-tête de demande d'ECP: de SP à ECP

Le bloc d'en-tête SOAP de demande d'ECP est utilisé pour convoyer des informations nécessaires à l'ECP pour traiter la demande d'authentification. Il est obligatoire et sa présence signale l'utilisation de ce profil. Il contient les éléments et attributs suivants:

– `SOAP-ENV:mustUnderstand` [Exigé]

  La valeur doit être 1 (vrai). Une faute SOAP doit être générée si le bloc d'en-tête d'ECP n'est pas compris.

– `SOAP-ENV:actor` [Exigé]

  La valeur doit être `http://schemas.xmlsoap.org/soap/actor/next`.

– `ProviderName` [Facultatif]

  Nom lisible par l'homme pour le fournisseur de service demandeur.

– `IsPassive` [Facultatif]

  Valeur booléenne. Si il est réglé à `vrai`, le fournisseur d'identité et le client lui-même ne doivent pas prendre le contrôle de l'interface d'utilisateur sur le producteur de la demande et interagir avec le principal d'une façon notable. Si aucune valeur n'est fournie, elle doit être `vraie` par défaut.

– `<saml:Issuer>` [Exigé]

  Cet élément doit contenir l'identifiant unique du fournisseur de service demandeur; l'attribut Format doit être omis ou avoir une valeur de `urn:oasis:names:tc:SAML:2.0:nameid-format:entity`.

– `<samlp:IDPList>` [Facultatif]

  Liste facultative des fournisseurs d'identité que le fournisseur de service reconnaît et à partir desquels l'ECP peut choisir de servir la demande.

11.4.2.4.3 Bloc d'en-tête RelayState d'ECP: SP à ECP

Le bloc d'en-tête SOAP RelayState d'ECP est utilisé pour convoyer les informations d'état provenant du fournisseur de service dont il aura besoin ultérieurement lors du traitement de la réponse de l'ECP. Il est facultatif, mais s'il est utilisé, l'ECP doit inclure un bloc d'en-tête identique dans la réponse à l'étape 5 dans la Figure 11-2. Il contient les attributs suivants:

NOTE (informative) – PE27 (voir OASIS PE:2006) suggère de remplacer étape 5 par étape 7 dans le texte ci-dessus.

– `SOAP-ENV:mustUnderstand` [Exigé]

  La valeur doit être 1 (vrai). Une faute SOAP doit être générée si le bloc d'en-tête n'est pas compris.

– `SOAP-ENV:actor` [Exigé]

  La valeur doit être `http://schemas.xmlsoap.org/soap/actor/next`.

Le contenu de l'élément bloc d'en-tête est une chaîne qui contient des informations d'état créées par le demandeur. S'il est fourni, l'ECP doit inclure la même valeur dans un bloc d'en-tête RelayState lorsqu'il répond au fournisseur de service à l'étape 5. La valeur de la chaîne ne doit pas excéder 80 octets de long et devrait être protégée en intégrité par le demandeur indépendamment de toutes autres protections qui pourraient ou non exister durant la transmission du message.

Ci-après figure un exemple de la demande d'authentification SOAP du fournisseur de service à l'ECP:

```
<SOAP-ENV:Envelope
       xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion"
       xmlns:samlp="urn:oasis:names:tc:SAML:2.0:protocol"
       xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Header>
    <paos:Request xmlns:paos="urn:liberty:paos:2003-08"
       responseConsumerURL="http://identity-service.example.com/abc"
       messageID="6c3a4f8b9c2d" SOAP-
ENV:actor="http://schemas.xmlsoap.org/soap/actor/next" SOAP-
ENV:mustUnderstand="1"
       service="urn:oasis:names:tc:SAML:2.0:profiles:SSO:ecp">
    </paos:Request>
    <ecp:Request xmlns:ecp="urn:oasis:names:tc:SAML:2.0:profiles:SSO:ecp"
       SOAP-ENV:mustUnderstand="1" SOAP-
ENV:actor="http://schemas.xmlsoap.org/soap/actor/next"
       ProviderName="Service Provider X" IsPassive="0">
      <saml:Issuer>https://ServiceProvider.example.com</saml:Issuer>
      <samlp:IDPList>
        <samlp:IDPEntry ProviderID="https://IdentityProvider.example.com"
            Name="Identity Provider X"
            Loc="https://IdentityProvider.example.com/saml2/sso"
        </samlp:IDPEntry>
        <samlp:GetComplete>
        https://ServiceProvider.example.com/idplist?id=604be136-fe91-441e-
afb8
        </samlp:GetComplete>
      </samlp:IDPList>
    </ecp:Request>
    <ecp:RelayState
xmlns:ecp="urn:oasis:names:tc:SAML:2.0:profiles:SSO:ecp"
       SOAP-ENV:mustUnderstand="1" SOAP-
ENV:actor="http://schemas.xmlsoap.org/soap/actor/next">
      ...
    </ecp:RelayState>
  </SOAP-ENV:Header>
  <SOAP-ENV:Body>
    <samlp:AuthnRequest> ... </samlp:AuthnRequest>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Comme noté ci-dessus, les blocs d'en-tête PAOS et ECP sont retirés du message SOAP par l'ECP avant que la demande d'authentification ne soit transmise au fournisseur d'identité. Un exemple de demande d'authentification de l'ECP au fournisseur d'identité est donné ci-après:

```
<SOAP-ENV:Envelope xmlns:SOAP-
ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:samlp="urn:oasis:names:tc:SAML:2.0:protocol"
```

```
  <SOAP-ENV:Body>
    <samlp:AuthnRequest> ... </samlp:AuthnRequest>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

11.4.2.4.4 Bloc d'en-tête de réponse d'ECP: de IdP à ECP

Le bloc d'en-tête SOAP de réponse d'ECP doit être utilisé sur la réponse provenant du fournisseur d'identité à l'ECP. Il contient les attributs suivants:

– `SOAP-ENV:mustUnderstand` [Exigé]

  La valeur doit être 1 (vrai). Une faute SOAP doit être générée si le bloc d'en-tête ECP n'est pas compris.

– `SOAP-ENV:actor` [Exigé]

  La valeur doit être `http://schemas.xmlsoap.org/soap/actor/next`.

– `AssertionConsumerServiceURL` [Exigé]

  Réglé par le fournisseur d'identité sur la base du message `<AuthnRequest>` ou des métadonnées du fournisseur de service obtenues par le fournisseur d'identité.

L'ECP doit confirmer que cette valeur correspond à la valeur d'ECP obtenue dans le `responseConsumerURL` dans le bloc d'en-tête SOAP de demande PAOS qu'il a reçu du fournisseur de service. Comme le `responseConsumerURL` peut être relatif et que le `AssertionConsumerServiceURL` est absolu, du traitement/normalisation peut être nécessaire.

Ce mécanisme est utilisé pour les besoins de la sécurité pour confirmer que la destination de réponse est correcte. Si les valeurs ne correspondent pas, l'ECP doit alors générer une réponse de faute SOAP au fournisseur de service et ne doit pas retourner la réponse SOAP.

L'en-tête SOAP de réponse d'ECP n'a pas d'élément contenu.

Ci-après figure un exemple d'une réponse d'IdP-à-ECP.

```
<SOAP-ENV:Envelope
       xmlns:ecp="urn:oasis:names:tc:SAML:2.0:profiles:SSO:ecp"
       xmlns:samlp="urn:oasis:names:tc:SAML:2.0:protocol"
       xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Header>
     <ecp:Response SOAP-ENV:mustUnderstand="1" SOAP-
ENV:actor="http://schemas.xmlsoap.org/soap/actor/next"
AssertionConsumerServiceURL="https://ServiceProvider.example.com/ecp_assert
ion_consumer"/>
  </SOAP-ENV:Header>
  <SOAP-ENV:Body>
    <samlp:Response> ... </samlp:Response>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

11.4.2.4.5 Bloc d'en-tête de réponse de PAOS: d'ECP à SP

Le bloc d'en-tête de réponse de PAOS inclut les attributs suivants:

– `SOAP-ENV:mustUnderstand` [Exigé]

  La valeur doit être 1 (vrai). Une faute SOAP doit être générée si le bloc d'en-tête de réponse de PAOS n'est pas compris.

– `SOAP-ENV:actor` [Exigé]

  La valeur doit être `http://schemas.xmlsoap.org/soap/actor/next`.

– `refToMessageID` [Facultatif]

  Permet la corrélation avec la demande de PAOS. Cet attribut facultatif (et le bloc d'en-tête dans sa totalité) doit être ajouté par l'ECP si la demande de PAOS correspondante spécifiait l'attribut `messageID`. La fonctionnalité équivalente est fournie dans SAML en utilisant la corrélation de `<AuthnRequest>` et de `<Response>`.

Le bloc d'en-tête de réponse de PAOS n'a pas d'élément contenu.

Ci-après figure un exemple de réponse d'ECP-à-SP.

```
<SOAP-ENV:Envelope
       xmlns:paos="urn:liberty:paos:2003-08"
       xmlns:samlp="urn:oasis:names:tc:SAML:2.0:protocol"
       xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Header>
    <paos:Response refToMessageID="6c3a4f8b9c2d" SOAP-
ENV:actor="http://schemas.xmlsoap.org/soap/actor/next/" SOAP-
ENV:mustUnderstand="1"/>
    <ecp:RelayState
xmlns:ecp="urn:oasis:names:tc:SAML:2.0:profiles:SSO:ecp"
       SOAP-ENV:mustUnderstand="1" SOAP-
ENV:actor="http://schemas.xmlsoap.org/soap/actor/next">
      ...
    </ecp:RelayState>
  </SOAP-ENV:Header>
  <SOAP-ENV:Body>
    <samlp:Response> ... </samlp:Response>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

11.4.2.5 Considérations sur la sécurité

Le message `<AuthnRequest>` devrait être signé. Selon les règles spécifiées par le profil de navigateur SSO, les assertions incluses dans la `<Response>` doivent être signées. La livraison de la réponse dans l'enveloppe SOAP via PAOS est essentiellement analogue à l'utilisation de la liaison HTTP POST et des contre-mesures de sécurité appropriées à cette liaison sont utilisées.

Les en-têtes SOAP devraient être protégés en intégrité, de la même façon qu'avec la sécurité de message SOAP ou qu'avec l'utilisation de TLS sur chaque échange HTTP avec le client.

Le fournisseur de service devrait être authentifié auprès de l'ECP, par exemple avec l'authentification TLS côté serveur.

L'ECP devrait être authentifié auprès du fournisseur d'identité, comme en maintenant une session authentifiée. Tout échange HTTP suivant la livraison du message `<AuthnRequest>` et avant que le fournisseur d'identité ne retourne une `<Response>` doit être associé de façon sécurisée à la demande d'origine.

NOTE (informative) – PE20 (voir OASIS PE:2006) suggère l'ajout d'un paragraphe pour exposer les considérations sur les métadonnées d'ECP:

Les règles spécifiées dans le profil SSO de navigateur au § 11 s'appliquent aussi ici. Précisément, l'élément de point d'extrémité indexé `<md:AssertionConsumerService>` avec une liaison de `urn:oasis:names:tc:SAML:2.0:bindings:PAOS`, peut être utilisé pour décrire la liaison prise en charge et la ou les localisations auxquelles un fournisseur d'identité peut envoyer des réponses à un fournisseur de service en utilisant ce profil. Et le point d'extrémité `<md:SingleSignOnService>` avec une liaison de `urn:oasis:names:tc:SAML:2.0:bindings:SOAP`, peut être utilisé pour décrire la liaison prise en charge et la ou les localisations auxquelles un fournisseur de service peut envoyer une demande à un fournisseur d'identité en utilisant ce profil.

11.4.3 Profil de découverte de fournisseur d'identité

Le présent paragraphe définit un profil par lequel un fournisseur de service peut découvrir quels fournisseurs d'identité utilise un principal avec le profil SSO de navigateur de la toile. Dans les développements qui ont plus d'un fournisseur d'identité, les fournisseurs de service ont besoin d'un moyen de découvrir quel ou quels fournisseurs d'identité utilise un principal. Le profil de découverte s'appuie sur un mouchard qui est écrit dans un domaine commun entre fournisseurs d'identité et fournisseurs de service dans un développement. Le domaine que prédétermine le développement est connu comme le domaine commun dans ce profil, et le mouchard qui contient la liste des fournisseurs d'identité est connu comme le mouchard de domaine commun.

La détermination des entités qui hébergent les serveurs de la toile dans le domaine commun est une question de développement et est en dehors du domaine d'application de ce profil.

NOTE (informative) – PE32 (voir OASIS PE:2006) suggère d'ajouter ce qui suit pour décrire les informations nécessaires:

Identification: urn:oasis:names:tc:SAML:2.0:profiles:SSO:idp-discovery

Informations de contact: security-services-comment@lists.oasis-open.org

11.4.3.1 Mouchard de domaine commun

Le nom du mouchard doit être "`_saml_idp`". Le format de la valeur du mouchard doit être un ensemble de une ou plusieurs valeurs d'URI codées en base64 séparées par un seul caractère d'espace. Chaque URI est l'identifiant unique d'un fournisseur d'identité, comme défini au § 7. L'ensemble final des valeurs est alors codé en URL.

Le service d'écriture de mouchard de domaine commun devrait ajouter l'identifiant unique de fournisseur d'identité à la liste. Si l'identifiant est déjà présent dans la liste, il peut le retirer et l'ajouter. L'intention est que la session de fournisseur d'identité la plus récemment établie soit la dernière de la liste.

Le mouchard doit être réglé avec un préfixe Path de "/". Le Domain doit être réglé à ".[common-domain]" où [common-domain] est le domaine commun établi au sein du développement à utiliser avec ce profil. Il doit y avoir une période d'apprentissage. Le mouchard doit être marqué comme sécurisé.

La syntaxe de mouchard devrait être conforme à la RFC 2965 de l'IETF. Le mouchard peut être seulement pour la session ou être persistant. Ce choix peut être fait au sein d'un développement, mais devrait s'appliquer uniformément à tous les fournisseurs d'identité du développement.

11.4.3.2 Réglage du mouchard de domaine commun

Après que le fournisseur d'identité a authentifié un principal, il peut régler le mouchard de domaine commun. Les moyens par lesquels le fournisseur d'identité règle le mouchard sont spécifiques de l'implémentation pour autant que le mouchard soit bien réglé avec les paramètres donnés ci-dessus. Une stratégie d'implémentation possible est présentée ci-dessous et devrait être considérée comme non normative. Le fournisseur d'identité peut:

- avoir précédemment établi un alias DNS et IP pour lui-même dans le domaine commun;
- rediriger l'agent d'utilisateur sur lui-même en utilisant l'alias DNS en utilisant un URL qui spécifie "https" comme schéma d'URL. La structure de l'URL est particulière à l'implémentation et peut inclure les informations de session nécessaires pour identifier l'agent d'utilisateur;
- régler le mouchard sur l'agent d'utilisateur de redirection en utilisant les paramètres spécifiés ci-dessus;
- rediriger l'agent d'utilisateur sur lui-même, ou si c'est approprié, sur le fournisseur de service.

11.4.3.3 Obtenir le mouchard de domaine commun

Lorsqu'un fournisseur de service a besoin de découvrir quels fournisseurs d'identité utilise un principal, il invoque un échange destiné à présenter le mouchard de domaine commun au fournisseur de service après qu'il a été lu par un serveur HTTP dans le domaine commun.

Si le fournisseur de service fait fonctionner le serveur HTTP dans le domaine commun ou si d'autres arrangements sont pris, le fournisseur de service peut utiliser le serveur HTTP dans le domaine commun pour relayer son `<AuthnRequest>` au fournisseur d'identité pour un processus optimisé de signature unique.

Les moyens spécifiques par lesquels le fournisseur de service lit le mouchard sont spécifiques de l'implémentation pour autant qu'ils soient capables de faire que l'agent d'utilisateur présente les mouchards qui ont été établis avec les paramètres donnés au § 11.4.3.1. Une stratégie d'implémentation possible est décrite ci-dessus et devrait être considérée comme non normative. De plus, elle peut être sous optimale pour certaines applications.

- Avoir précédemment établi un alias DNS et IP pour soi-même dans le domaine commun.
- Rediriger l'agent d'utilisateur sur lui-même en utilisant l'alias DNS à l'aide d'un URL spécifiant "https" comme schéma d'URL. La structure de l'URL est particulière à l'implémentation et peut inclure les informations de session nécessaires pour identifier l'agent d'utilisateur.
- Rediriger l'agent d'utilisateur sur lui-même, ou si approprié, sur le fournisseur d'identité.

11.4.4 Profil de terminaison de session unique

Une fois qu'un principal s'est authentifié à un fournisseur d'identité, l'entité d'authentification peut établir une session avec le principal (normalement au moyen d'un mouchard, de la réécriture d'URL, ou par quelque autre moyen spécifique de l'implémentation). Le fournisseur d'identité peut ultérieurement produire des assertions au fournisseur de services ou aux autres consommateurs d'assertions, sur la base de cet événement d'authentification; un consommateur d'assertions peut utiliser cela pour établir *sa propre* session avec le principal.

Dans une telle situation, le fournisseur d'identité peut agir comme autorité de session et les consommateurs d'assertions comme participants de session. Ultérieurement, le principal peut souhaiter terminer sa session avec un participant de session individuel, ou avec tous les participants de session dans une session donnée gérée par l'autorité de session. Le premier cas est considéré comme sortant du domaine d'application de la présente Recommandation. Le second peut cependant être satisfait en utilisant ce profil du protocole de terminaison de session unique de SAML (voir au § 11.4).

Un principal (ou un administrateur qui met fin à une session d'un principal) peut choisir de terminer sa session "globale" en contactant l'autorité de session, ou un participant individuel de session. Un fournisseur d'identité agissant comme autorité de session peut aussi *lui-même* agir comme un participant de session dans des situations dans lesquelles il est le consommateur d'assertions pour les assertions concernant ce principal d'un autre fournisseur d'identité.

Le profil permet au protocole d'être combiné avec une liaison synchrone, telle que la liaison SOAP, ou avec des liaisons de "canal frontal" asynchrones, telles que les liaisons HTTP Redirect, POST, ou Artifact. Une liaison de canal frontal peut être nécessaire, par exemple, dans les cas où un état de session d'un principal n'existe que dans un seul agent d'utilisateur sous la forme d'un mouchard et où une interaction directe entre l'agent d'utilisateur et le participant de session ou autorité de session est nécessaire. Comme il sera exposé ci-dessous, le participant de session devrait si possible utiliser une liaison de "canal frontal" lors de l'initialisation de ce profil, pour maximiser la probabilité que l'autorité de session puisse propager la terminaison de session avec succès à tous les participants.

11.4.4.1 Informations requises

Identification: `urn:oasis:names:tc:SAML:2.0:profiles:SSO:logout`

Informations de contact: security-services-comment@lists.oasis-open.org

Description: donnée ci-dessous

Mises à jour: aucune

11.4.4.2 Description générale du profil

La Figure 11-3 illustre le schéma de base de réalisation de la terminaison de session unique:

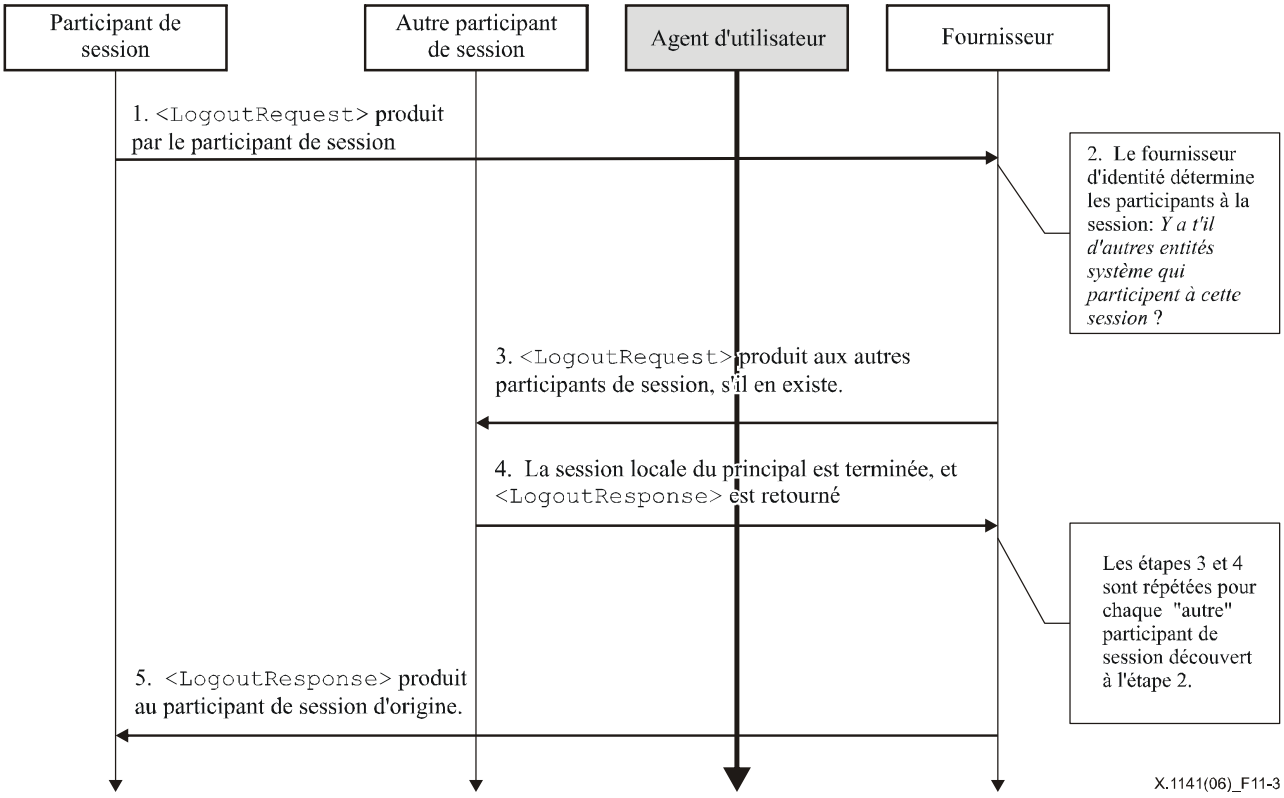

Figure 11-3/X.1141 – Schéma de réalisation de la terminaison de session unique

L'agent d'utilisation en grisé illustre que cet échange de messages peut passer à travers l'agent d'utilisateur ou peut être un échange direct entre entités système, selon la liaison SAML utilisée pour implémenter le profil.

Les étapes suivantes sont décrites par le profil. Au sein d'une étape individuelle, il peut y avoir un ou plusieurs échanges de messages réels selon la liaison utilisée pour cette étape et autre comportement dépendant de l'implémentation.

1) <LogoutRequest> produit par le participant de session au fournisseur d'identité

A l'étape 1, le participant de session initie une seule terminaison de session et termine la ou les sessions d'un principal en envoyant un message `<LogoutRequest>` au fournisseur d'identité dont il a reçu l'assertion d'authentification correspondante. La demande peut être envoyée directement au fournisseur d'identité ou envoyée indirectement à travers l'agent d'utilisateur.

2) Le fournisseur d'identité détermine les participants de session

A l'étape 2, le fournisseur d'identité utilise le contenu du message `<LogoutRequest>` (ou s'il initie la terminaison de session lui-même, quelque autre mécanisme) pour déterminer la ou les sessions qui se terminent. S'il n'y pas d'autre participant de session, le profil passe à l'étape 5. Autrement, les étapes 3 et 4 sont répétées pour chaque participant de session identifié.

3) <LogoutRequest> produit par le fournisseur d'identité au participant/autorité de session

A l'étape 3, le fournisseur d'identité produit un message `<LogoutRequest>` se rapportant à une ou plusieurs des sessions qui se terminent à un participant de session ou autorité de session. La demande peut être envoyée directement à l'entité ou envoyée indirectement à travers l'agent d'utilisateur (si c'est cohérent avec la forme de la demande à l'étape 1).

4) Le participant/autorité de session produit un <LogoutResponse> au fournisseur d'identité

A l'étape 4, un participant de session ou autorité de session termine le ou les sessions du principal selon les instructions de la demande (si possible) et retourne un `<LogoutResponse>` au fournisseur d'identité. La réponse peut être retournée directement au fournisseur d'identité ou indirectement à travers l'agent d'utilisateur (si c'est cohérent avec la forme de la demande à l'étape 3).

5) Le fournisseur d'identité produit un <LogoutResponse> au participant de session

A l'étape 5, le fournisseur d'identité produit un message `<LogoutResponse>` au participant de session à l'origine de la demande. La réponse peut être retournée directement au participant de session ou indirectement à travers l'agent d'utilisateur (si c'est cohérent avec la forme de la demande à l'étape 1).

Un fournisseur d'identité (agissant comme autorité de session) peut initier ce profil à l'étape 2 et produire un `<LogoutRequest>` à tous les participants de session, et aussi en sautant l'étape 5.

11.4.4.3 Description de profil

Si le profil est initialisé par un participant de session, commencer au § 11.4.4.3.1. S'il est initialisé par le fournisseur d'identité, commencer au § 11.4.4.3.2. Dans les descriptions suivantes, on se réfère à ce qui suit:

– **Service de terminaison de session unique**

C'est le point d'extrémité de protocole de terminaison de session unique chez un fournisseur d'identité ou participant de session auquel les messages `<LogoutRequest>` ou `<LogoutResponse>` (ou un artifice les représentant) sont délivrés. Les mêmes points d'extrémité ou des points d'extrémité différents peuvent être utilisés dans les demandes et les réponses.

11.4.4.3.1 <LogoutRequest> produit par le participant de session au fournisseur d'identité

Si le profil de terminaison de session est initié par un participant de session, il examine la ou les assertions d'authentification reçues qui appartiennent à la ou aux sessions qui se terminent, et collecte la ou les valeurs de `SessionIndex` reçues du fournisseur d'identité. Si plusieurs fournisseurs d'identité sont impliqués, le profil doit alors être répété de façon indépendante pour chacun d'eux.

Pour initialiser le profil, le participant de session produit un message `<LogoutRequest>` au point d'extrémité de demande de service de terminaison de session unique du fournisseur d'identité qui contient un ou plusieurs éléments `<SessionIndex>` applicables. Au moins un élément doit être inclus. Les métadonnées peuvent être utilisées pour déterminer la localisation de ce point d'extrémité et les liaisons prises en charge par le fournisseur d'identité.

Liaisons asynchrones (canal frontal)

Le participant de session devrait (si l'agent d'utilisateur du principal est présent) utiliser une liaison asynchrone, telle qu'une liaison HTTP Redirect, POST, ou Artifact (voir au § 10), pour envoyer la demande au fournisseur d'identité à travers l'agent d'utilisateur. Le fournisseur d'identité devrait alors propager tout message de terminaison de session nécessaire aux participants de session supplémentaires en utilisant une liaison synchrone ou asynchrone. L'utilisation d'une liaison asynchrone pour la demande d'origine est préférée parce qu'elle donne au fournisseur d'identité les meilleurs chances de propagation réussie de la terminaison de session aux autres participants de session durant l'étape 3 au § 11.4.4.2.

Si la liaison HTTP Redirect ou POST est utilisée, le message `<LogoutRequest>` est alors délivré au fournisseur d'identité à cette étape. Si la liaison HTTP Artifact est utilisée, le profil de résolution d'artifice défini au § 11.4.6 est utilisé par le fournisseur d'identité, qui effectue un rappel au participant de session pour restituer le message `<LogoutRequest>`, en utilisant par exemple, la liaison SOAP.

Il est recommandé que les échanges HTTP à cette étape soient faits sur TLS 1.0 pour conserver la confidentialité et l'intégrité du message. Le message `<LogoutRequest>` doit être signé si la liaison HTTP POST ou Redirect est utilisée.

La liaison HTTP Artifact, si elle est utilisée, fournit aussi un moyen de remplacement d'authentification du producteur de la demande lorsque l'artifice est déréférencé.

Chacune de ces liaisons fournit un mécanisme RelayState que le participant de session peut utiliser pour associer l'échange de profils à la demande d'origine. Le participant de session devrait révéler aussi peu que possible d'informations dans la valeur de `RelayState`, à moins que l'utilisation du profil n'exige pas de telles mesures de confidentialité.

Liaisons synchrones (canal de retour)

Autrement, le participant de session peut utiliser une liaison synchrone, telle que la liaison SOAP (voir au § 10), pour envoyer la demande directement au fournisseur d'identité. Le fournisseur d'identité devrait alors propager tous les messages de terminaison de session nécessaires aux participants de session supplémentaires en utilisant une liaison synchrone. Le demandeur doit s'authentifier auprès du fournisseur d'identité, soit en signant le `<LogoutRequest>`, soit en utilisant tout autre mécanisme pris en charge par la liaison.

Les règles spécifiques de profil pour le contenu du message `<LogoutRequest>` sont incluses au § 11.4.4.4.1.

11.4.4.3.2 Le fournisseur d'identité détermine les participants de session

Si le profil de terminaison de session est initié par un fournisseur d'identité, ou à réception d'un message `<LogoutRequest>` valide, le fournisseur d'identité traite la demande, il doit examiner les éléments ID et `<SessionIndex>` et déterminer l'ensemble des sessions à terminer.

Le fournisseur d'identité suit ensuite les étapes 3 et 4 dans la Figure 11-3 pour chaque entité participant à la ou aux sessions qui se terminent, autre que le participant de session de la demande d'origine (s'il en est), comme décrit au § 8.2.7.

11.4.4.3.3 <LogoutRequest> produit par le fournisseur d'identité au participant/autorité de session

Pour propager la terminaison de session, le fournisseur d'identité produit son propre `<LogoutRequest>` à une autorité de session ou participant à une session en cours de terminaison. La demande est envoyée, en utilisant une liaison SAML cohérente avec les capacités du répondant et la disponibilité de l'agent d'utilisateur, au fournisseur d'identité.

En général, la liaison avec laquelle la demande d'origine a été reçue à l'étape 1 dans la Figure 11-3 ne dicte pas quelle liaison peut être utilisée à cette étape, excepté comme noté à l'étape 1. L'utilisation d'une liaison synchrone court-circuitant l'agent d'utilisateur contraint le fournisseur d'identité à utiliser une liaison similaire pour propager une demande supplémentaire.

Les règles spécifiques de profil pour le contenu du message `<LogoutRequest>` sont incluses au § 11.4.4.4.1.

11.4.4.3.4 Le participant/autorité de session produit <LogoutResponse> au fournisseur d'identité

Le participant/autorité de session doit traiter le message `<LogoutRequest>` comme défini au § 8.2.7. Après le traitement du message ou en rencontrant une erreur, l'entité doit produire un message `<LogoutResponse>` contenant un code d'état approprié au fournisseur d'identité demandeur pour terminer l'échange de protocole SAML.

Liaisons synchrones (canal de retour)

Si le fournisseur d'identité a utilisé une liaison synchrone, comme la liaison SOAP (voir au § 10), la réponse est retournée directement pour terminer la communication synchrone. Le répondant doit s'authentifier auprès du fournisseur d'identité demandeur, soit en signant le `<LogoutResponse>`, soit en utilisant un autre mécanisme pris en charge par la liaison.

Liaisons asynchrones (canal frontal)

Si le fournisseur d'identité a utilisé une liaison asynchrone, comme une liaison HTTP Redirect, POST, ou Artifact (voir au § 10), le `<LogoutResponse>` (ou son artifice) est retourné à travers l'agent d'utilisateur au point d'extrémité de réponse de service de terminaison de session unique du fournisseur d'identité. Des métadonnées peuvent être utilisées pour déterminer la localisation de ce point d'extrémité et les liaisons prises en charge par le fournisseur d'identité. Toute liaison asynchrone prise en charge par les deux entités peut être utilisée.

Si la liaison HTTP Redirect ou POST est utilisée, le message `<LogoutResponse>` est délivré au fournisseur d'identité à cette étape. Si la liaison HTTP Artifact est utilisée, le profil de résolution d'artifice défini au § 11.4.6 est utilisé par le fournisseur d'identité, qui fait un rappel de l'entité répondante pour restituer le message `<LogoutResponse>`, en utilisant par exemple, la liaison SOAP.

Il est recommandé que les échanges HTTP à cette étape soient faits sur TLS 1.0 pour conserver la confidentialité et l'intégrité du message. Le message `<LogoutResponse>` doit être signé si la liaison HTTP POST ou Redirect est

utilisée. La liaison HTTP Artifact, si elle est utilisée, fournit aussi un moyen de remplacement pour authentifier le producteur de la réponse lorsque l'artifice est déréférencé.

Les règles spécifiques du profil pour le contenu du message <LogoutResponse> sont incluses au § 11.4.4.4.2.

11.4.4.3.5 Le fournisseur d'identité produit un <LogoutResponse> au participant de session

Après le traitement du <LogoutRequest> du participant de session original, comme décrit aux étapes précédentes, le fournisseur d'identité doit répondre à la demande d'origine par un <LogoutResponse> contenant un code d'état approprié pour terminer l'échange de protocole SAML.

La réponse est envoyée au participant de session original, en utilisant une liaison SAML cohérente avec la liaison utilisée dans la demande d'origine, avec les capacités du répondant, et la disponibilité de l'agent d'utilisateur envers le fournisseur d'identité. En supposant qu'une liaison asynchrone a été utilisée à l'étape 1 dans la Figure 11-3, toute liaison prise en charge par les deux entités peut être utilisée.

Les règles spécifiques du profil pour le contenu du message <LogoutResponse> sont incluses au § 11.4.4.4.2.

11.4.4.4 Utilisation du protocole de terminaison de session unique

Ce paragraphe décrit l'utilisation de <LogoutRequest> et <LogoutResponse>.

11.4.4.4.1 Utilisation de <LogoutRequest>

L'élément <Issuer> doit être présent et doit contenir l'identifiant unique de l'entité demandeuse; l'attibut Format doit être omis ou avoir une valeur de urn:oasis:names:tc:SAML:2.0:nameid-format:entity.

Le demandeur doit s'authentifier auprès du répondant et s'assurer de l'intégrité du message, soit en signant le message, soit en utilisant un mécanisme spécifique de la liaison.

Le principal doit être identifié dans la demande en utilisant un identifiant qui **corresponde fortement** à l'identifiant dans l'assertion d'authentification que le demandeur a produite ou reçue en ce qui concerne la session à terminer, selon les règles de correspondance définies au § 8.2.7.

Si le demandeur est un participant de session, il doit inclure au moins un élément <SessionIndex> dans la demande. Si le demandeur est une autorité de session (ou s'il agit en son nom), il peut alors omettre de tels éléments pour indiquer la terminaison à toutes les sessions applicables du principal.

> NOTE (informative) – PE38 (voir OASIS PE:2006) précise l'alinéa ci-dessus comme suit:
>
> Si le demandeur est un participant de session, il doit inclure au moins un élément <SessionIndex> dans la demande. (D'après le § 11.4, le participant de session reçoit toujours un attribut SessionIndex dans les éléments <saml:AuthnStatement> qu'il reçoit pour initier la session). Si le demandeur est une autorité de session (ou agit en son nom), il peut alors omettre de tels éléments pour indiquer la terminaison de toutes les sessions applicables du principal.

11.4.4.4.2 Utilisation de <LogoutResponse>

L'élément <Issuer> doit être présent et doit contenir l'identifiant unique de l'entité qui répond; l'attribut Format doit être omis ou avoir une valeur de urn:oasis:names:tc:SAML:2.0:nameid-format:entity.

Le répondant doit s'authentifier auprès du demandeur et s'assurer de l'intégrité du message, soit en signant le message, soit en utilisant un mécanisme spécifique de la liaison.

11.4.4.5 Utilisation des métadonnées

L'élément point d'extrémité, <md:SingleLogoutService>, décrit les liaisons prises en charge et la ou les localisations auxquelles une entité peut envoyer demandes et réponses en utilisant ce profil. Un demandeur, s'il chiffre l'identifiant du principal, peut utiliser l'élément <md:KeyDescriptor> du répondant avec un attribut d'utilisation de chiffrement pour déterminer un algorithme de chiffrement approprié et les réglages à utiliser, ainsi qu'une clé publique à utiliser à la livraison d'une clé de chiffrement brute.

11.4.5 Profil de gestion d'identifiant de nom

Dans le scénario pris en charge par le profil de gestion d'identifiant de nom, un fournisseur d'identité a échangé une forme d'identifiant persistant pour un principal avec un fournisseur de service, leur permettant de partager un identifiant commun pour une certaine durée. Ensuite, le fournisseur d'identité peut souhaiter notifier au fournisseur de service un changement de format et/ou de valeur qu'il utilisera pour identifier le même principal à l'avenir. Autrement, le fournisseur de service peut souhaiter attacher son propre "alias" au principal afin de s'assurer que le fournisseur d'identité l'incluera à l'avenir lors de communications avec lui au sujet du principal. Finalement, un des fournisseurs

peut souhaiter informer l'autre qu'il ne produira plus ou n'acceptera plus de messages utilisant un identifiant particulier. Pour implémenter ces scénarios, on utilise un profil de protocole de gestion d'identifiant de nom SAML.

> NOTE (informative) – PE12 (voir OASIS PE:2006) suggère de réécrire la seconde phrase de l'alinéa ci-dessus comme suit:
>
> Ensuite, le fournisseur d'identité peut souhaiter notifier au fournisseur de service un changement de la valeur qu'il utilisera pour identifier le même principal à l'avenir.

Le profil permet au protocole d'être combiné avec une liaison synchrone, telle qu'une liaison SOAP, ou avec des liaisons asynchrones de "canal frontal", telles que les liaisons HTTP Redirect, POST, ou Artifact. Une liaison de canal frontal peut être nécessaire, par exemple, dans des cas où une interaction directe entre l'agent d'utilisateur et le fournisseur répondant est nécessaire pour effectuer le changement.

11.4.5.1 Informations requises

Identification: `urn:oasis:names:tc:SAML:2.0:profiles:SSO:nameid-mgmt`

Informations de contact: security-services-comment@lists.oasis-open.org

Description: donnée ci-dessous

Mises à jour: aucune.

11.4.5.2 Aperçu général du profil

La Figure 11-4 illustre le schéma de base pour le profil de gestion d'identifiant de nom.

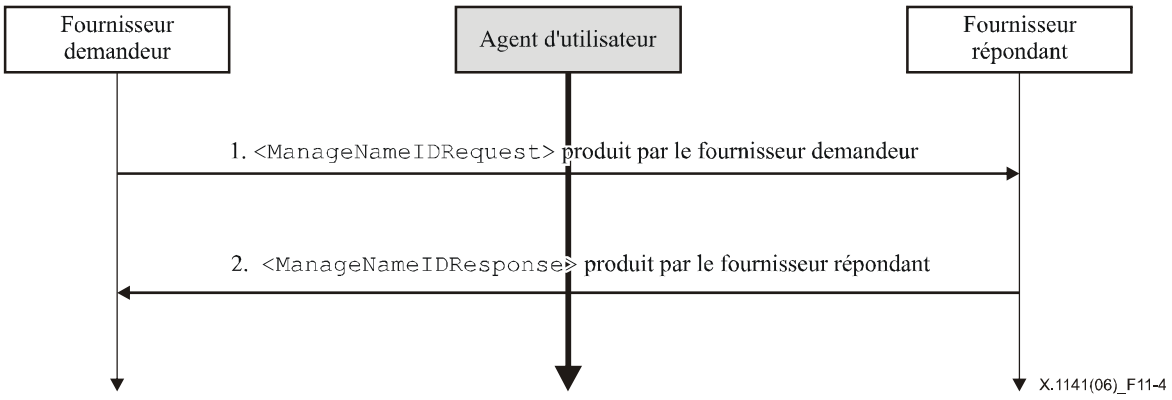

Figure 11-4/X.1141 – Profil de gestion d'identifiant de nom

L'agent d'utilisateur en grisé illustre le fait que l'échange de messages peut passer à travers l'agent d'utilisateur ou peut être un échange direct entre entités système, selon la liaison SAML utilisée pour implémenter le profil.

Les étapes suivantes sont décrites par le profil. Au sein de chaque étape, il peut y avoir un ou plusieurs échanges réels de messages selon la liaison utilisée pour cette étape et d'autres selon le comportement de l'implémentation.

1) **<ManageNameIDRequest> produit par le fournisseur d'identité/service demandeur**

   A l'étape 1, un fournisseur d'identité ou de service initie le profil en envoyant un message `<ManageNameIDRequest>` à un autre fournisseur qu'il souhaite informer d'un changement. La demande peut être envoyée directement au fournisseur répondant ou envoyée indirectement à travers l'agent d'utilisateur.

2) **<ManageNameIDResponse>produit par le fournisseur d'identité/service répondant**

   A l'étape 2, le fournisseur répondant (après traitement de la demande) produit un message `<ManageNameIDResponse>` au fournisseur demandeur d'origine. La réponse peut être retournée directement au fournisseur demandeur ou indirectement à travers l'agent d'utilisateur (si c'est cohérent avec la forme de la demande à l'étape 1).

11.4.5.3 Description de profil

Dans les descriptions ci-dessous, on se réfère à ce qui suit:

Service de gestion d'identifiant de nom

C'est le point d'extrémité de protocole de gestion d'identifiant de nom chez un fournisseur d'identité ou de service auquel les messages `<ManageNameIDRequest>` ou `<ManageNameIDResponse>` (ou un artifice qui les représente) sont délivrés. Des points d'extrémité identiques ou différents peuvent être utilisés pour les demandes et les réponses.

11.4.5.3.1 <ManageNameIDRequest> produit par le fournisseur d'identité/service demandeur

Pour initier le profil, le fournisseur demandeur produit un message `<ManageNameIDRequest>` à un point d'extrémité de demande de service de gestion d'identifiant de nom d'un autre fournisseur. Des métadonnées peuvent être utilisées pour déterminer la localisation de ce point d'extrémité et les liaisons prises en charge par le fournisseur répondant.

– **Liaisons synchrones (canal de retour)**

  Le fournisseur demandeur peut utiliser une liaison synchrone, telle qu'une liaison SOAP (voir au § 10), pour envoyer la demande directement à l'autre fournisseur. Le demandeur doit s'authentifier auprès de l'autre fournisseur, en signant le `<ManageNameIDRequest>` ou en utilisant tout autre mécanisme pris en charge par la liaison.

– **Liaisons asynchrones (canal frontal)**

  Autrement, le fournisseur demandeur peut (si l'agent d'utiilsateur du principal est présent) utiliser une liaison asynchrone, telle qu'une liaison HTTP Redirect, POST, ou Artifact (voir au § 10), pour envoyer la demande à l'autre fournisseur à travers l'agent d'utilisateur.

Si la liaison HTTP Redirect ou POST est utilisée, le message `<ManageNameIDRequest>` est délivré à l'autre fournisseur à cette étape. Si la liaison HTTP Artifact est utilisée, le profil de résolution d'artifice défini au § 11.4.6 est utilisé par l'autre fournisseur, qui fait un rappel au fournisseur demandeur pour restituer le message `<ManageNameIDRequest>`, en utilisant par exemple la liaison SOAP.

Il est recommandé que les échanges HTTP à cette étape soient faits sur TLS 1.0 pour conserver la confidentialité et l'intégrité du message. Le message `<ManageNameIDRequest>` doit être signé si la liaison HTTP POST ou Redirect est utilisée. La liaison HTTP Artifact, si elle est utilisée, fournit aussi un moyen de remplacement d'authentification du producteur de la demande lorsque l'artifice est déréférencé.

Chacune de ces liaisons fournit un mécanisme RelayState que le fournisseur demandeur peut utiliser pour associer l'échange de profils à la demande d'origine. Le fournisseur demandeur devrait révéler le moins d'informations possible dans la valeur `RelayState` sauf si l'utilisation du profil n'exige pas de telles mesures de confidentialité.

Les règles spécifiques du profil pour le contenu du message `<ManageNameIDRequest>` figurent au § 11.4.5.4.1.

11.4.5.3.2 <ManageNameIDResponse> produit par le fournisseur d'identité/service répondant

Le receveur doit traiter le message `<ManageNameIDRequest>`. Après le traitement du message ou en rencontrant une erreur, le receveur doit produire un message `<ManageNameIDResponse>` contenant un code d'état approprié au fournisseur demandeur pour terminer l'échange de protocole SAML.

– **Liaisons synchrones (canal de retour)**

  Si le fournisseur demandeur a utilisé une liaison synchrone, telle qu'une liaison SOAP (voir au § 10), la réponse est retournée directement pour terminer la communication synchrone. Le répondant doit s'authentifier auprès du fournisseur demandeur, soit en signant le `<ManageNameIDResponse>`, soit en utilisant tout autre mécanisme pris en charge par la liaison.

– **Liaisons asynchrones (canal frontal)**

  Si le fournisseur demandeur a utilisé une liaison asynchrone, telle qu'une liaison HTTP Redirect, POST, ou Artifact (voir au § 10), le `<ManageNameIDResponse>` (ou son artifice) est alors retourné à travers l'agent d'utilisateur au point d'extrémité de réponse de gestion d'identifiant de nom au fournisseur demandeur. Des métadonnées peuvent être utilisées pour déterminer la localisation de ce point d'extrémité et les liaisons prises en charge par le fournisseur demandeur. Toute liaison prise en charge par les deux entités peut être utilisée.

Si la liaison HTTP Redirect ou POST est utilisée, le message `<ManageNameIDResponse>` est alors délivré au fournisseur demandeur à cette étape. Si la liaison HTTP Artifact est utilisée, le profil de résolution d'artifice défini au § 11.4.6 est utilisé par le fournisseur demandeur, qui fait un rappel au fournisseur répondant pour restituer le message `<ManageNameIDResponse>`, en utilisant par exemple la liaison SOAP.

Il est recommandé que les échanges HTTP à cette étape soient faits sur TLS 1.0 pour conserver la confidentialité et l'intégrité du message. Le message `<ManageNameIDResponse>` doit être signé si la liaison HTTP POST ou Redirect

est utilisée. La liaison HTTP Artifact, si elle est utilisée, fournit aussi un moyen de remplacement de l'authentification du producteur de la réponse lorsque l'artifice est déréférencé.

Les règles spécifiques du profil pour le contenu du message `<ManageNameIDResponse>` figurent au § 11.4.5.4.2.

11.4.5.4 Utilisation du protocole de gestion d'identifiant de nom

Le présent paragraphe couvre l'utilisation de `ManageNameIDRequest` et `ManageNameIDResponse`.

11.4.5.4.1 Utilisation de <ManageNameIDRequest>

L'élément `<Issuer>` doit être présent et doit contenir l'identifiant unique de l'entité demandeuse; l'attribut `Format` doit être omis ou avoir une valeur de `urn:oasis:names:tc:SAML:2.0:nameid-format:entity`.

Le demandeur doit s'authentifier auprès du répondant et assurer l'intégrité du message, soit en signant le message, soit en utilisant un mécanisme spécifique de la liaison.

11.4.5.4.2 Utilisation de <ManageNameIDResponse>

L'élément `<Issuer>` doit être présent et doit contenir l'identifiant unique de l'entité répondante; l'attribut `Format` doit être omis ou avoir une valeur de `urn:oasis:names:tc:SAML:2.0:nameid-format:entity`.

Le répondant doit s'authentifier auprès du demandeur et assurer l'intégrité du message, soit en signant le message, soit en utilisant un mécanisme spécifique de la liaison.

11.4.5.5 Utilisation des métadonnées

L'élément point d'extrémité `<md:ManageNameIDService>` décrit les liaisons prises en charge et la ou les localisations auxquelles une entité peut envoyer des demandes et réponses en utilisant ce profil. Un demandeur, s'il chiffre l'identifiant du principal, peut utiliser l'élément `<md:KeyDescriptor>` avec un attribut d'utilisation de chiffrement pour déterminer un algorithme de chiffrement approprié et des réglages à utiliser, avec une clé publique à utiliser à la livraison d'une clé de chiffrement brute.

11.4.6 Profil de résolution d'artifice

Le paragraphe 10 définit un protocole de résolution d'artifice ou de déréférencement d'un artifice SAML en un message de protocole correspondant. La liaison HTTP Artifact (voir le § 10) accentue ce mécanisme pour passer les messages de protocole SAML par référence. Ce profil décrit l'utilisation de ce protocole avec une liaison synchrone, telle que la liaison SOAP définie au § 10.

11.4.6.1 Informations requises

Identification: `urn:oasis:names:tc:SAML:2.0:profiles:artifact`

Informations de contact: security-services-comment@lists.oasis-open.org

Description: donnée ci-dessous

Mises à jour: aucune

11.4.6.2 Aperçu général du profil

L'échange de messages et les règles de base du traitement qui gouvernent ce profil sont bien définies au § 8 qui définit les messages à échanger, en combinaison avec la liaison utilisée pour échanger les messages. Le paragraphe 10 définit la liaison cet échange de messages avec SOAP V1.1. Sauf mention spécifique dans la présente Recommandation, toutes les exigences définie dans ces spécifications s'appliquent.

La Figure 11-5 illustre le schéma de base pour le profil de résolution d'artifice.

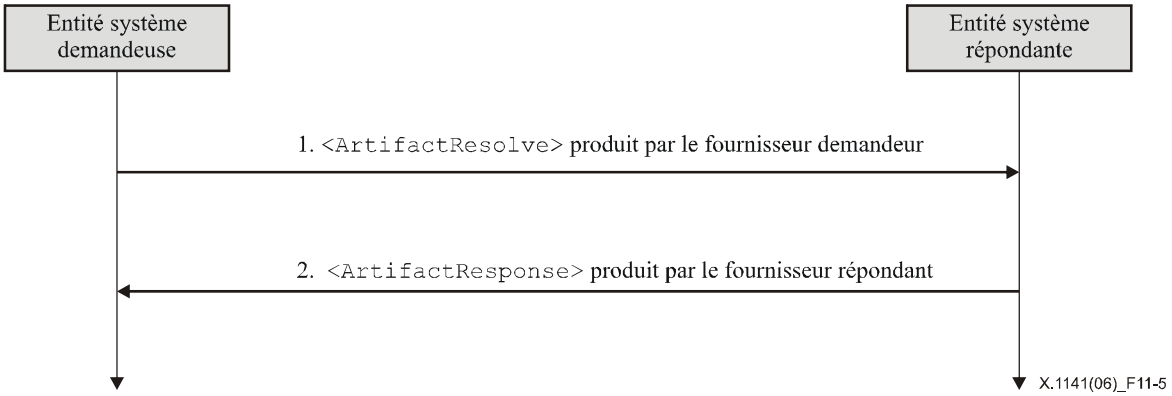

Figure 11-5/X.1141 – Schéma de base de profil de résolution d'artifice

Le profil décit les étapes suivantes.

1) <ArtifactResolve> produit par l'entité demandeuse

A l'étape 1, un demandeur initie le profil en envoyant un message `<ArtifactResolve>` à un producteur d'artifice.

2) <ArtifactResponse> produit par l'entité répondante

A l'étape 2, le répondant (après traitement de la demande) produit un message `<ArtifactResponse>` au demandeur.

11.4.6.3 Description de profil

Dans les descriptions ci-dessous, on se réfère à:

– **Service de résolution d'artifice**

C'est le point d'extrémité de protocole de résolution d'artifice, chez un producteur d'artifice, auquel les messages `<ArtifactResolve>` sont livrés.

11.4.6.3.1 <ArtifactResolve> produit par l'entité demandeuse

Pour initier le profil, un demandeur qui a reçu un artifice et déterminé son producteur en utilisant le `SourceID`, envoie un message `<ArtifactResolve>` contenant l'artifice à un point d'extrémité de service de résolution d'artifice de producteur d'artifice. Des métadonnées peuvent être utilisées pour déterminer la localisation de ce point d'extrémité et les liaisons prises en charge par le producteur d'artifice.

Le demandeur doit utiliser une liaison synchrone, comme la liaison SOAP (voir le § 10), pour envoyer la demande directement au producteur d'artifice. Le demandeur devrait s'authentifier auprès du répondant, soit en signant le message `<ArtifactResolve>`, soit en utilisant tout autre mécanisme pris en charge par la liaison. Des profils spécifiques qui utilisent la liaison HTTP Artifact peuvent imposer des exigences supplémentaires comme l'authentification obligatoire.

Les règles spécifiques du profil pour le contenu du message `<ArtifactResolve>` figurent au § 11.4.6.4.1.

11.4.6.3.2 <ArtifactResponse> produit par l'entité répondante

Le producteur d'artifice doit traiter le message `<ArtifactResolve>` comme défini au § 8. Après le traitement du message ou en rencontrant une erreur, le producteur d'artifice doit retourner un message `<ArtifactResponse>` contenant un code d'état approprié au demandeur pour terminer l'échange de protocole SAML. S'il réussit, le message de protocole SAML déréférencé correspondant à l'artifice devra aussi être inclus.

Le répondant doit s'authentifier auprès du demandeur, soit en signant le `<ArtifactResponse>`, soit en utilisant tout autre mécanisme pris en charge par la liaison.

Les règles spécifiques du profil pour le contenu du message `<ArtifactResponse>` figurent au § 11.4.6.4.2.

11.4.6.4 Utilisation du protocole de résolution d'artifice

Ce paragraphe couvre l'utilisation de `ArtifactResolve` et `ArtifactResponse`.

11.4.6.4.1 Utilisation de <ArtifactResolve>

L'élément `<Issuer>` doit être présent et doit contenir l'identifiant unique de l'entité demandeuse; l'attribut `Format` doit être omis ou avoir une valeur de `urn:oasis:names:tc:SAML:2.0:nameid-format:entity`.

Le demandeur devrait s'authentifier auprès du répondant et assurer l'intégrité du message, soit en signant le message, soit en utilisant un mécanisme spécifique de la liaison. Des profils spécifiques qui utilisent la liaison HTTP Artifact peuvent imposer des exigences supplémentaires comme l'authentification obligatoire.

11.4.6.4.2 Utilisation de <ArtifactResponse>

L'élément `<Issuer>` doit être présent et doit contenir l'identifiant unique du producteur d'artifice; l'attribut `Format` doit être omis ou avoir une valeur de `urn:oasis:names:tc:SAML:2.0:nameid-format:entity`.

Le répondant doit s'authentifier auprès du demandeur et assurer l'intégrité du message, soit en signant le message, soit en utilisant un mécanisme spécifique de la liaison.

11.4.6.5 Utilisation des métadonnées

Le paragraphe 9 définit un élément de point d'extrémité indexé, `<md:ArtifactResolutionService>`, pour décrire les liaisons prises en charge et la ou les localisations auxquelles un demandeur peut envoyer une demande en utilisant ce profil. L'attribut `index` est utilisé pour distinguer les points d'extrémité possibles qui peuvent être spécifiés par référence dans le champ `EndpointIndex` de l'artifice.

11.4.7 Profil d'interrogation/demande d'assertion

Le paragraphe 10 définit un protocole pour demander les assertions existantes par référence ou par interrogation sur la base d'un sujet et de critères supplémentaires spécifiques d'une déclaration. Ce profil décrit l'utilisation de ce protocole avec une liaison synchrone, comme la liaison SOAP définie au § 10.

11.4.7.1 Informations requises

Identification: `urn:oasis:names:tc:SAML:2.0:profiles:query`

Informations de contact: security-services-comment@lists.oasis-open.org

Description: donnée ci-dessous.

Mises à jour: aucune.

11.4.7.2 Aperçu général du profil

L'échange de messages et les règles de base du traitement qui gouvernent ce profil sont bien définies au § 8 qui définit les messages à échanger, en combinaison avec la liaison utilisée pour échanger les messages. Le paragraphe 10 définit la liaison de l'échange de messages avec SOAP V1.1. Sauf mention spécifique, toutes les exigences définies dans ces spécifications s'appliquent.

La Figure 11-6 illustre le schéma de base du profil d'interrogation/demande.

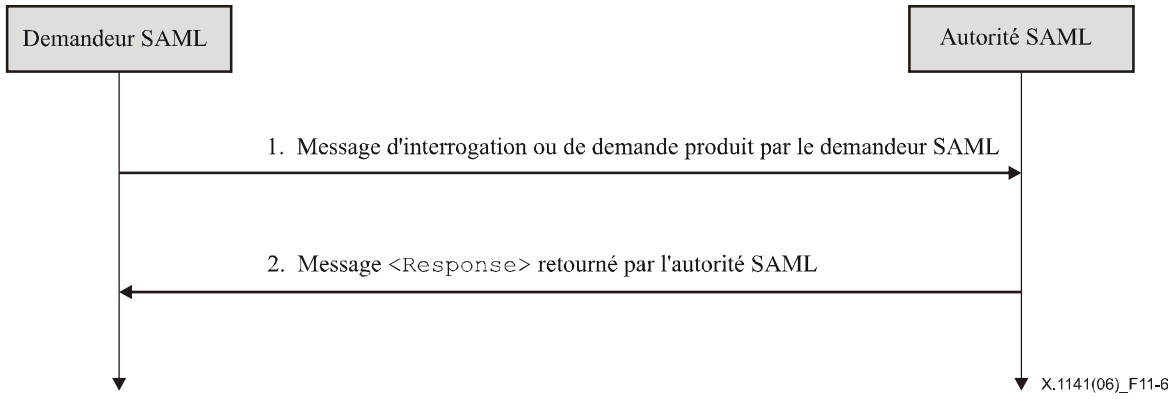

Figure 11-6/X.1141 – Schéma de base du profil d'interrogation/demande

Le profil décrit les étapes suivantes:

1) Interrogation/demande produite par le demandeur SAML

A l'étape 1, un demandeur SAML initie le profil en envoyant un message `<AssertionIDRequest>`, `<SubjectQuery>`, `<AuthnQuery>`, `<AttributeQuery>`, ou `<AuthzDecisionQuery>` à une autorité SAML.

2) <Response>produite par l'autorité SAML

A l'étape 2, l'autorité SAML répondante (après traitement de l'interrogation ou de la demande) produit un message `<Response>` au demandeur SAML.

11.4.7.3 Description de profil

Dans les descriptions ci-dessous, on se réfère à ce qui suit:

– **Service d'interrogation/demande**

C'est le point d'extrémité de protocole d'interrogation/demande chez une autorité SAML auquel les interrogations ou les messages `<AssertionIDRequest>` sont délivrés.

11.4.7.3.1 Interrogation/demande produite par le demandeur SAML

Pour initier le profil, un demandeur SAML produit un message `<AssertionIDRequest>`, `<SubjectQuery>`, `<AuthnQuery>`, `<AttributeQuery>`, ou `<AuthzDecisionQuery>` à un point d'extrémité de service d'interrogation/demande d'une autorité SAML. Les métadonnées peuvent être utilisées pour déterminer la localisation de ce point d'extrémité et les liaisons prises en charge par l'autorité SAML.

Le demandeur SAML doit utiliser une liaison synchrone, comme la liaison SOAP (voir le § 10), pour envoyer la demande directement au fournisseur d'identité. Le demandeur devrait s'authentifier auprès de l'autorité SAML soit en signant le message, soit en utilisant tout autre mécanisme pris en charge par la liaison.

Les règles spécifiques du profil pour le contenu des divers messages figurent au § 11.4.7.4.1.

11.4.7.3.2 <Response> produit par l'autorité SAML

L'autorité SAML doit traiter l'interrogation ou le message de demande comme défini au § 8. Après le traitement du message ou en rencontrant une erreur, l'autorité SAML doit retourner un message `<Response>` contenant un code d'état approprié au demandeur SAML pour terminer l'échange de protocole SAML. Si la demande réussit à localiser une ou plusieurs assertions qui correspondent, elles seront aussi incluses dans la réponse.

Le répondant devrait s'authentifier auprès du demandeur, soit en signant la `<Response>`, soit en utilisant tout autre mécanisme pris en charge par la liaison.

Les règles spécifiques du profil pour le contenu du message `<Response>` figurent au § 11.4.7.4.2.

11.4.7.4 Utilisation du protocole d'interrogation/demande

Le présent paragraphe définit le point d'extrémité de protocole d'interrogation/demande, chez une autorité SAML, auquel les messages d'interrogation sont délivrés.

11.4.7.4.1 Utilisation de l'interrogation/demande

L'élément `<Issuer>` doit être présent.

Le demandeur devrait s'authentifier auprès du répondant et assurer l'intégrité du message, soit en signant le message, soit en utilisant un mécanisme spécifique de la liaison.

11.4.7.4.2 Utilisation de <Response>

L'élément `<Issuer>` doit être présent et doit contenir l'identifiant unique de l'autorité SAML répondante; l'attribut Format doit être omis ou avoir une valeur de `urn:oasis:names:tc:SAML:2.0:nameid-format:entity`. Cela ne doit pas nécessairement correspondre à l'élément `<Issuer>` dans la ou les assertions retournées.

Le répondant devrait s'authentifier auprès du demandeur et assurer l'intégrité du message, soit en signant le message, soit en utilisant un mécanisme spécifique de la liaison.

11.4.7.5 Utilisation des métadonnées

Le paragraphe 9 définit plusieurs éléments de point d'extrémité, `<md:AssertionIDRequestService>`, `<md:AuthnQueryService>`, `<md:AttributeService>`, et `<md:AuthzService>`, pour décrire les liaisons

acceptées et la ou les localisations auxquelles un demandeur peut envoyer les demandes ou interrogations en utilisant ce profil.

L'autorité SAML, si elle chiffre les assertions résultantes ou les contenus d'assertion pour une entité particulière, peut utiliser l'élément `<md:KeyDescriptor>` de cette entité avec un attribut d'utilisation du chiffrement pour déterminer un algorithme de chiffrement approprié et les réglages à utiliser, avec une clé publique à utiliser pour la livraison d'une clé de chiffrement brute.

Les divers descripteurs de rôle peuvent contenir des éléments `<md:NameIDFormat>`, `<md:AttributeProfile>`, et `<saml:Attribute>` (selon ce qui est applicable) pour indiquer la capacité générale à prendre en charge des formats d'identifiant de nom particuliers, des profils d'attribut, ou des attributs et valeurs spécifiques. La capacité à prendre en charge une de ces caractéristiques durant une demande donnée dépend de la politique et est à la discrétion de l'autorité.

11.4.8 Profil de mappage d'identifiant de nom

Le paragraphe 8.2.6 définit un protocole de mappage d'identifiant de nom pour mapper l'identifiant de nom d'un principal en un identifiant de nom différent pour le même principal. Ce profil décrit l'utilisation de ce protocole avec une liaison synchrone, comme la liaison SOAP définie au § 10, et des lignes directrices supplémentaires pour la protection de la confidentialité du principal avec le chiffrement et en limitant l'utilisation de l'identifiant mappé.

11.4.8.1 Informations requises

Identification: `urn:oasis:names:tc:SAML:2.0:profiles:nameidmapping`

Informations de contact: security-services-comment@lists.oasis-open.org

Description: donnée ci-dessous.

Mises à jour: aucune.

11.4.8.2 Aperçu général du profil

L'échange de messages et les règles de base du traitement qui gouvernent ce profil sont bien définies au § 8 qui définit les messages à échanger, en combinaison avec la liaison utilisée pour échanger les messages. Le paragraphe 10 définit la liaison de l'échange de messages avec SOAP V1.1. Sauf mention spécifique, toutes les exigences définies dans ces spécifications s'appliquent.

La Figure 11-7 illustre le schéma de base pour le profil de transposition d'identifiant de nom.

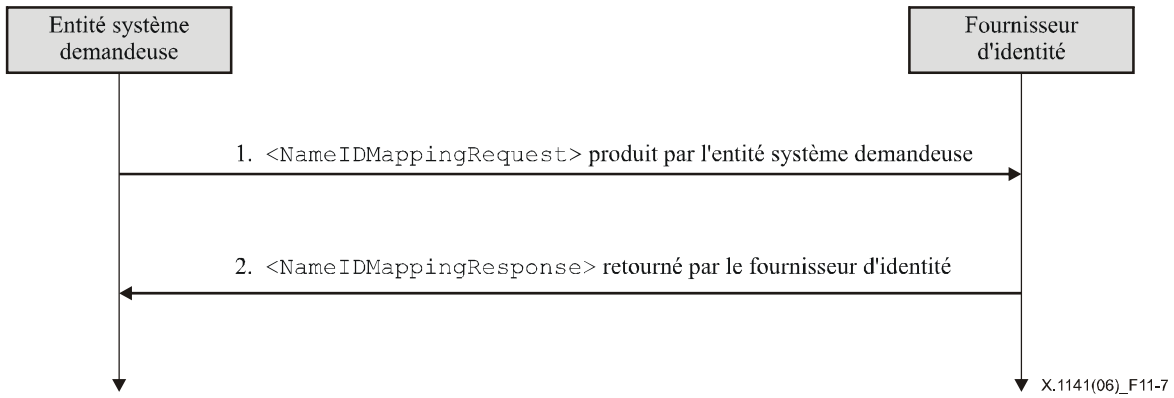

Figure 11-7/X.1141 – Schéma de base de profil d'identifiant de nom

Le profil décrit les étapes suivantes:

1) <NameIDMappingRequest> produit par l'entité demandeuse

A l'étape 1, un demandeur initie le profil en envoyant un message `<NameIDMappingRequest>` à un fournisseur d'identité.

2) <NameIDMappingResponse> produit par le fournisseur d'identité

A l'étape 2, le fournisseur d'identité répondant (après traitement de la demande) produit un message `<NameIDMappingResponse>` au demandeur.

11.4.8.3 Description de profil

Le présent paragraphe utilise le service de mappage d'identifiant de nom, qui est le point d'extrémité de protocole de mappage d'identifiant de nom, chez un fournisseur d'identité, auquel les messages `<NameIDMappingRequest>` sont livrés.

11.4.8.3.1 <NameIDMappingRequest> produit par l'entité demandeuse

Pour initier le profil, un demandeur produit un message `<NameIDMappingRequest>` à un point d'extrémité de service de mappage d'identifiant de nom d'un fournisseur d'identité. Les métadonnées peuvent être utilisées pour déterminer la localisation de ce point d'extrémité et les liaisons prises en charge par le fournisseur d'identité.

Le demandeur doit utiliser une liaison synchrone, comme la liaison SOAP (voir § 10), pour envoyer la demande directement au fournisseur d'identité. Le demandeur doit s'authentifier auprès du fournisseur d'identité, soit en signant le `<NameIDMappingRequest>`, soit en utilisant tout autre mécanisme pris en charge par la liaison.

Les règles spécifiques du profil pour le contenu du message `<NameIDMappingRequest>` figurent au § 11.4.8.4.1.

11.4.8.3.2 <NameIDMappingResponse> produit par le fournisseur d'identité

Le fournisseur d'identité doit traiter le message `<ManageNameIDRequest>` comme défini au § 8. Après le traitement du message ou en rencontrant une erreur, le fournisseur d'identité doit retourner un message `<NameIDMappingResponse>` contenant un code d'état approprié au demandeur pour terminer l'échange de protocole SAML.

Le répondant doit s'authentifier auprès du demandeur, soit en signant le `<NameIDMappingResponse>`, soit en utilisant tout autre mécanisme pris en charge par la liaison.

Les règles spécifiques du profil pour le contenu du message `<NameIDMappingResponse>` figurent au § 11.4.8.4.2.

11.4.8.4 Utilisation du protocole de mappage d'identifiant de nom

Le paragraphe 8 définit un protocole de mappage d'identifiant de nom pour le mappage d'un identifiant de nom d'un principal en un identifiant de nom différent pour le même principal. Le présent paragraphe décrit l'utilisation de ce protocole et des lignes directrices supplémentaires pour la protection de la confidentialité du principal comme la limitation de l'utilisation de l'identifiant mappé.

11.4.8.4.1 Utilisation de <NameIDMappingRequest>

L'élément `<Issuer>` doit être présent.

Le demandeur doit s'authentifier auprès du répondant et assurer l'intégrité du message, soit en signant le message, soit en utilisant un mécanisme spécifique de la liaison.

11.4.8.4.2 Utilisation de <NameIDMappingResponse>

L'élément `<Issuer>` doit être présent et doit contenir l'identifiant unique du fournisseur d'identité répondant; l'attribut `Format` doit être omis ou avoir une valeur de `urn:oasis:names:tc:SAML:2.0:nameid-format:entity`.

Le répondant doit s'authentifier auprès du demandeur et assurer l'intégrité du message, soit en signant le message, soit en utilisant un mécanisme spécifique de la liaison.

Le paragraphe 2.2.3 de chiffrement du W3C, définit l'utilisation du chiffrement pour appliquer la confidentialité à un identifiant de nom. Dans la plupart des cas, le fournisseur d'identité devrait chiffrer l'identifiant de nom mappé qu'il retourne au demandeur pour protéger la confidentialité du principal. Le demandeur peut extraire l'élément `<EncryptedID>` et le placer dans les messages de protocole ou assertions suivants.

Limitation de l'utilisation de l'identifiant mappé

Des restrictions supplémentaires à l'utilisation de l'identifiant résultant peuvent être appliquées par le fournisseur d'identité en retournant l'identifiant de nom mappé sous la forme d'un `<Assertion>` contenant l'identifiant dans son `<Subject>` mais sans aucune déclaration. L'assertion est alors chiffrée et le résultat est utilisé comme l'élément `<EncryptedData>` dans le `<EncryptedID>` retourné au demandeur. L'assertion peut inclure un élément `<Conditions>` pour en limiter l'utilisation, comme défini au § 8, comme une contrainte limitée en temps ou une utilisation par des consommateurs d'assertions spécifiques, et doit être signée pour une protection de l'intégrité.

11.4.8.5 Utilisation des métadonnées

Le présent paragraphe définit un élément de point d'extrémité, `<md:NameIDMappingService>`, pour décrire les liaisons prises en charge et la ou les localisations auxquelles un demandeur peut envoyer une demande en utilisant ce profil.

Le fournisseur d'identité, s'il chiffre l'identifiant résultant pour une entité particulière, peut utiliser cet élément `<md:KeyDescriptor>` d'entité avec un attribut d'utilisation de chiffrement pour déterminer un algorithme de chiffrement approprié et les réglages à utiliser, avec une clé publique à utiliser pour la livraison d'une clé de chiffrement brute.

11.4.9 Profils d'attribut SAML

Les profils d'attribut fournissent les définitions nécessaires pour limiter l'expression des attributs SAML lorsqu'on traite de types particuliers d'informations d'attribut ou lors d'interactions avec des systèmes externes qui requièrent d'être plus strictes. Le présent paragraphe spécifie le profil d'attribut de base SAML, le profil X.500/LDAP et les profils UUID et le profil XACML.

11.4.9.1 Profil d'attribut de base

Le profil d'attribut `Basic` (de base) spécifie la dénomination simplifiée, mais non unique, des attributs SAML ainsi que les valeurs d'attribut fondées sur les types de données de Datatypes incorporés du W3C, éliminant le besoin d'"un schéma d'extension pour valider la syntaxe.

Informations requises

Identification: `urn:oasis:names:tc:SAML:2.0:profiles:attribute:basic`

Informations de contact: security-services-comment@lists.oasis-open.org

Description: donnée ci-dessous

Mises à jour: aucune.

Dénomination d'attribut SAML

L'attribut XML `NameFormat` dans les éléments `<Attribute>` doit être `urn:oasis:names:tc:SAML:2.0:attrname-format:basic`.

L'attribut XML `Name` doit adhérer aux règles spécifiées pour ce format, comme défini au § 8.

– **Comparaison de nom d'attribut**

  Deux éléments `<Attribute>` se réfèrent au même attribut SAML si et seulement si les valeurs de leurs attributs XML Name sont égales (dans le sens décrit au § 8).

Attributs XML spécifiques de profil

Aucun attribut XML supplémentaire n'est défini pour être utilisé avec l'élément `<Attribute>`.

Valeurs d'attribut SAML

Le type de schéma du contenu de l'élément `<AttributeValue>` doit être tiré d'un des types définis à l'Annexe A. L'attribut `xsi:type` doit être présent et être à la valeur appropriée.

Exemple

```
<saml:Attribute NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-
format:basic"
              Name="FirstName">
      <saml:AttributeValue xsi:type="xs:string">By-
Tor</saml:AttributeValue>
</saml:Attribute>
```

11.4.9.2 Profil d'attribut X.500/LDAP

Les répertoires fondés sur les Recommandations UIT-T de la série X.500 et sur la RFC 3377 de l'IETF sont largement répandus. Le schéma de répertoire est utilisé pour modéliser les informations à mémoriser dans ces répertoires. En particulier, dans X.500, les définitions de type d'attribut sont utilisées pour spécifier la syntaxe et d'autres caractéristiques des attributs, unité de base du stockage des informations dans un répertoire (la présente Recommandation s'y réfère comme à des "attributs de répertoire"). Le types d'attribut de répertoire sont définis sous forme de schémas dans les spécifications X.500 et LDAP elle-mêmes, de schémas dans d'autres documents publics

(comme le schéma inetOrgperson (voir la RFC 2798 de l'IETF)), et de schémas définis pour des besoins privés. Dans tous ces cas, il est utile aux développeurs de tirer parti de ces types d'attributs de répertoire dans le contexte des déclarations d'attributs SAML, sans avoir à créer manuellement des définitions d'attribut spécifiques de SAML pour eux, et de le faire de façon interopérable.

Le profil d'attribut X.500/LDAP définit une convention commune pour la dénomination et la représentation de tels attributs lorsqu'ils sont exprimés comme attributs SAML.

Informations requises

Identification: `urn:oasis:names:tc:SAML:2.0:profiles:attribute:X500` (c'est aussi l'espace de nom cible alloué dans le schéma correspondant de profil X.500/LDAP à l'Annexe A).

Informations de contact: security-services-comment@lists.oasis-open.org

Description: donnée ci-dessous

Mises à jour: aucune.

Dénomination d'attribut SAML

L'attribut XML `NameFormat` dans les éléments `<Attribute>` doit être `urn:oasis:names:tc:SAML:2.0:attrname-format:uri`.

Pour construire les noms d'attribut, on utilise l'espace de nom `oid` d'URN décrit dans la RFC 3061 de l'IETF. Dans cette approche l'attribut XML Name se fonde sur l'identifiant d'objet alloué au type d'attribut de répertoire.

Exemple:

`urn:oid:2.5.4.3`

Comme les procédures de X.500 exigent que chaque type d'attribut soit identifié par un identifiant d'objet unique, ce schéma de dénomination assure que les noms d'attributs SAML dérivés sont sans ambiguïté.

Pour les besoins du lecteur humain, il peut aussi être nécessaire pour certaines applications de porter une chaîne de nom facultative avec l'URN OID (comme défini dans la RFC 3061 de l'IETF). L'attribut XML facultatif `FriendlyName` (défini au § 8) peut être utilisé à cette fin. Si la définition du type d'attribut de répertoire inclut un ou plusieurs descripteurs (diminutifs) pour le type d'attribut, la valeur `FriendlyName`, si elle est présente, devrait être un des descripteurs définis.

Deux éléments `<Attribute>` se réfèrent au même attribut SAML si et seulement si leurs valeurs d'attribut XML `Name` sont égales au sens de la RFC 3061 de l'IETF. L'attribut `FriendlyName` ne joue aucun rôle dans la comparaison.

Attributs XML spécifiques de profil

Aucun attribut XML supplémentaire n'est défini pour être utilisé avec l'élément `<Attribute>`.

Valeurs d'attribut SAML

Les définitions de type d'attribut de répertoire à utiliser dans les répertoires X.500 spécifient la syntaxe de l'attribut en utilisant ASN.1. Pour l'utilisation dans LDAP, les définitions d'attribut de répertoire incluent en plus une syntaxe LDAP qui spécifie comment les valeurs d'attribut ou d'assertion se conformant à la syntaxe doivent être représentées lorsqu'elles sont transférées dans le protocole LDAP (connu sous le nom de codage spécifique LDAP). Le codage spécifique LDAP produit habituellement des caractères Unicode en format UTF-8. Le présent profil d'attribut SAML spécifie la forme de valeurs d'attribut SAML pour les seuls attributs de répertoire qui ont des syntaxes LDAP. Des extensions à venir de ce profil pourront définir des formats de valeur d'attribut pour les attributs de répertoire dont la syntaxe spécifie d'autres codages.

Pour représenter les règles de codage utilisées pour une valeur d'attribut particulière, l'élément `<AttributeValue>` doit contenir un attribut XML nommé `Encoding` défini dans l'espace de nom XML `urn:oasis:names:tc:SAML:2.0:profiles:attribute:X500`.

Pour tout attribut de répertoire avec une syntaxe dont le codage spécifique de LDAP produit comme valeurs exclusivement des chaînes de caractères UTF-8, la valeur d'attribut SAML est codée aussi simplement que la chaîne UTF-8 elle-même, comme le contenu de l'élément `<AttributeValue>`, sans espace blanc supplémentaire. Dans de tels cas, l'attribut XML xsi:type doit être réglé à `xs:string`. L'attribut XML `Encoding` (de codage) spécifique de profil est fourni avec une valeur de LDAP.

La liste de certaines des syntaxes d'attribut LDAP (et des OID associés) auxquelles ceci s'applique est:

Description de type d'attribut	1.3.6.1.4.1.1466.115.121.1.3
Chaîne binaire	1.3.6.1.4.1.1466.115.121.1.6

Booléen	1.3.6.1.4.1.1466.115.121.1.7
Chaîne de pays	1.3.6.1.4.1.1466.115.121.1.11
DN	1.3.6.1.4.1.1466.115.121.1.12
Chaîne de répertoire	1.3.6.1.4.1.1466.115.121.1.15
Numéro téléphonique de télécopie	1.3.6.1.4.1.1466.115.121.1.22
Temps généralisé	1.3.6.1.4.1.1466.115.121.1.24
Chaîne IA5	1.3.6.1.4.1.1466.115.121.1.26
ENTIER	1.3.6.1.4.1.1466.115.121.1.27
Description de syntaxe LDAP	1.3.6.1.4.1.1466.115.121.1.54
Description de règle de correspondance	1.3.6.1.4.1.1466.115.121.1.30
Description d'utilisation de règle de correspondance	1.3.6.1.4.1.1466.115.121.1.31
Nom et UID facultatif	1.3.6.1.4.1.1466.115.121.1.34
Description de forme de nom	1.3.6.1.4.1.1466.115.121.1.35
Chaîne numérique	1.3.6.1.4.1.1466.115.121.1.36
Description de classe d'objets	1.3.6.1.4.1.1466.115.121.1.37
Chaîne d'octet	1.3.6.1.4.1.1466.115.121.1.40
OID	1.3.6.1.4.1.1466.115.121.1.38
Autre adresse électronique	1.3.6.1.4.1.1466.115.121.1.39
Adresse postale	1.3.6.1.4.1.1466.115.121.1.41
Adresse de présentation	1.3.6.1.4.1.1466.115.121.1.43
Chaîne imprimable	1.3.6.1.4.1.1466.115.121.1.44
Sous-chaîne d'assertion	1.3.6.1.4.1.1466.115.121.1.58
Numéro de téléphone	1.3.6.1.4.1.1466.115.121.1.50
Temps UTC	1.3.6.1.4.1.1466.115.121.1.53

Pour toutes les autres syntaxes LDAP, la valeur d'attribut est codée, comme le contenu de l'élément `<AttributeValue>`, en codant en base64 la valeur d'attribut LDAP codée en chaîne d'octets ASN.1 environnante. L'attribut XML `xsi:type` doit être réglé à **xs:base64Binary**. L'attribut XML `Encoding` spécifique du profil est fourni, avec une valeur de "`LDAP`".

Lors de la comparaison de l'égalité des valeurs d'attribut SAML, les régles de correspondance spécifiées pour le type d'attribut de répertoire correspondant doivent être observées (sensibilité à la casse, par exemple).

Schéma spécifique du profil

La liste de schémas suivante montre la définition de l'attribut XML spécifique de profil `Encoding` (voir l'Annexe A):

```
<schema
    targetNamespace="urn:oasis:names:tc:SAML:2.0:profiles:attribute:X500"
    xmlns="http://www.w3.org/2001/XMLSchema"
    elementFormDefault="unqualified"
    attributeFormDefault="unqualified"
    blockDefault="substitution"
    version="2.0">
    <annotation>
        <documentation>
            Document identifier: saml-schema-x500-2.0
            Location: http://docs.oasis-open.org/security/saml/v2.0/
            Revision history:
              V2.0 (March, 2005):
                Custom schema for X.500 attribute profile, first published
in SAML 2.0.
        </documentation>
    </annotation>
    <attribute name="Encoding" type="string"/>
</schema>
```

Exemple

Ci-après figure un exemple de mappage de l'attribut de répertoire "givenName", qui représente le premier nom du sujet d'asserrtion SAML. Son identifiant d'objet est {joint-iso-itu-t(2) ds(5) attributeType(4) givenName(42)} et sa syntaxe LDAP est Directory String.

```
<saml:Attribute
xmlns:x500="urn:oasis:names:tc:SAML:2.0:profiles:attribute:X500"
             NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:uri"
             Name="urn:oid:2.5.4.42" FriendlyName="givenName">
      <saml:AttributeValue xsi:type="xs:string"
             x500:Encoding="LDAP">Steven</saml:AttributeValue>
</saml:Attribute>
```

11.4.9.3 Profil d'attribut UUID

Le profil d'attribut UUID normalise l'expression des valeurs d'UUID comme noms et valeurs d'attributs SAML. Il s'applique lorsque le système source de l'attribut identifie un attribut ou sa valeur avec un UUID.

Informations requises

Identification: `urn:oasis:names:tc:SAML:2.0:profiles:attribute:UUID`

Informations de contact: security-services-comment@lists.oasis-open.org

Description: donnée ci-dessous

Mises à jour: aucune.

Dénomination d'attribut SAML

L'attribut XML `NameFormat` dans les éléments `<Attribute>` doit être `urn:oasis:names:tc:SAML:2.0:attrname-format:uri`.

Si la représentation sous-jacente du nom de l'attribut est un UUID, l'espace de nom uuid d'URN décrit dans la Rec. UIT-T X.667 est alors utilisé. Dans cette approche, l'attribut XML Name se fonde sur la forme d'URN de l'UUID sous-jacent qui identifie l'attribut.

Exemple:

```
urn:uuid:f81d4fae-7dec-11d0-a765-00a0c91e6bf6
```

Si la représentation sous-jacente du nom de l'attribut n'est pas un UUID, toute forme d'URI peut alors être utilisée dans l'attribut XML Name.

Pour les besoins de la lecture par l'homme, il peut aussi être nécessaire pour certaines applications de porter un nom de chaîne facultatif avec l'URI. L'attribut XML facultatif `FriendlyName` peut être utilisé à cette fin.

Deux éléments `<Attribute>` se réfèrent au même attribut SAML si et seulement si leurs valeurs d'attribut XML `Name` sont égales au sens de la Rec. UIT-T X.667. L'attribut `FriendlyName` ne joue aucun rôle dans la comparaison.

Attributs XML spécifiques du profil

Aucun attribut XML supplémentaire n'est défini pour être utilisé dans l'élément `<Attribute>`.

Valeurs d'attribut SAML

Dans les cas où la valeur de l'attribut est aussi un UUID, la même syntaxe d'URN que décrite ci-dessus doit être utilisée pour exprimer la valeur dans l'élément `<AttributeValue>`. L'attribut XML `xsi:type` doit être réglé à **xs:anyURI**.

Si la valeur de l'attribut n'est pas un UUID, il n'y a alors aucune restriction sur l'utilisation de l'élément `<AttributeValue>`.

Exemple

Ci-après figure un exemple d'un attribut de répertoire étendu DCE, réglé à "pre_auth_req", qui a un UUID bien connu de 6c9d0ec8-dd2d-11cc-abdd-080009353559 et est en valeurs d'entiers.

```
<saml:Attribute NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-
format:uri"
             Name="urn:uuid:6c9d0ec8-dd2d-11cc-abdd-080009353559"
             FriendlyName="pre_auth_req">
```

```
      <saml:AttributeValue xsi:type="xs:integer">1</saml:AttributeValue>
</saml:Attribute>
```

11.4.9.4 Profil d'attribut XACML

Les assertions d'attribut SAML peuvent être utilisées en entrées aux décisions d'autorisation faites conformément à la Rec. UIT-T X.1142. Comme le format d'attribut SAML diffère du format d'attribut XACML, un mappage doit être effectué. Le profil d'attribut XACML facilite ce mappage en normalisant la dénomination, la syntaxe de valeur, et les métadonnées d'attribut supplémentaires. Les attributs SAML générés en conformité avec ce profil peuvent être mappés automatiquement en attributs XACML et utilisés en entrée des décisions d'autorisation XACML.

Informations requises

Identification: `urn:oasis:names:tc:SAML:2.0:profiles:attribute:XACML` (c'est aussi l'espace de nom cible alloué dans le schéma de profil XACML correspondant à l'Annexe A).

Informations de contact: security-services-comment@lists.oasis-open.org

Description: donnée ci-dessous.

Mises à jour: aucune.

Dénomination d'attribut SAML

L'attribut XML `NameFormat` dans les éléments `<Attribute>` doit être `urn:oasis:names:tc:SAML:2.0:attrname-format:uri`.

L'attribut XML `Name` doit adhérer aux règles spécifiées pour ce format, comme défini au § 8.

Pour les besoins de la lecture par l'homme, il peut aussi être nécessaire pour certaines applications de porter un nom de chaîne facultatif avec l'URN OID. L'attribut XML facultatif `FriendlyName` (défini au § 8) peut être utilisé à cette fin, mais il n'est pas traduisible en un attribut XACML équivalent.

Deux éléments `<Attribute>` se réfèrent au même attribut SAML si et seulement si leurs valeurs d'attribut XML Name sont égales dans une comparaison binaire. L'attribut `FriendlyName` ne joue aucun rôle dans la comparaison.

Attributs XML spécifiques du profil

XACML exige que chaque attribut porte un type de données explicite. Pour fournir cette valeur de type de données, un nouvel attribut XML à valeur d'URI appelé `DataType` est défini dans l'espace de nom XML `urn:oasis:names:tc:SAML:2.0:profiles:attribute:XACML`.

Les éléments SAML `<Attribute>` qui se conforment à ce profil doivent inclure l'attribut qualifié d'espace de nom `DataType`, ou la valeur est présumée être `http://www.w3.org/2001/XMLSchema#string`.

Si des valeurs non standards sont utilisées, chaque PDP XACML qui consommera des attributs SAML mappés avec des valeurs de `DataType` non standards doit être étendu pour accepter les nouveaux types de données.

Valeurs d'attribut SAML

La syntaxe du contenu de l'élément `<AttributeValue>` doit correspondre au type de données exprimé dans l'attribut XML `DataType` spécifique du profil qui apparait dans l'élément parent `<Attribute>`. Pour les types de données correspondants aux types définis au § 8, l'attribut XML xsi:type devrait aussi être utilisé sur le ou les éléments `<AttributeValue>`.

Schéma spécifique du profil

La liste de schémas suivante montre comment l'attribut `XML` spécifique du profil `DataType` est défini (Annexe A):

```
<schema
    targetNamespace="urn:oasis:names:tc:SAML:2.0:profiles:attribute:XACML"
    xmlns="http://www.w3.org/2001/XMLSchema"
    elementFormDefault="unqualified"
    attributeFormDefault="unqualified"
    blockDefault="substitution"
    version="2.0">
    <annotation>
        <documentation>
            Document identifier: saml-schema-xacml-2.0
            Location: http://docs.oasis-open.org/security/saml/v2.0/
            Revision history:
            V2.0 (March, 2005):
```

```
              Custom schema for XACML attribute profile, first published in
SAML 2.0.
        </documentation>
    </annotation>
    <attribute name="DataType" type="anyURI"/>
</schema>
```

Exemple

Ci-après figure un exemple de mappage de l'attribut LDAP/X.500 "givenName", qui représente le premier nom du sujet d'assertion SAML. Il illustre aussi qu'un seul attribut SAML peut se conformer à plusieurs profils d'attribut lorsqu'ils sont compatibles entre eux.

```
<saml:Attribute
xmlns:xacmlprof="urn:oasis:names:tc:SAML:2.0:profiles:attribute:XACML"
       xmlns:ldapprof="urn:oasis:names:tc:SAML:2.0:profiles:attribute:LDAP"
              xacmlprof:DataType="http://www.w3.org/2001/XMLSchema#string"
              ldapprof:Encoding="LDAP"
              NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:uri"
              Name="urn:oid:2.5.4.42" FriendlyName="givenName">
       <saml:AttributeValue xsi:type="xs:string">By-
Tor</saml:AttributeValue>
</saml:Attribute>
```

NOTE (informative) – PE39 (voir OASIS PE:2006) précise l'exemple ci-dessus de la façon suivante:

```
<saml:Attribute
  xmlns:xacmlprof="urn:oasis:names:tc:SAML:2.0:profiles:attribute:XACML"
      xmlns:ldapprof="urn:oasis:names:tc:SAML:2.0:profiles:AttributeValue:LDAP"
           xacmlprof:DataType="http://www.w3.org/2001/XMLSchema#string"
           ldapprof:Encoding="LDAP"
           NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:uri"
           Name="urn:oid:2.5.4.42" FriendlyName="givenName">
      <saml:AttributeValue xsi:type="xs:string">By-Tor</saml:AttributeValue>
</saml:Attribute>
```

12 Contexte d'authentification SAML

La présente Recommandation définit une syntaxe pour la définition des déclarations de contexte d'authentification et une liste initiale de classes de contexte d'authentification.

12.1 Concepts de contexte d'authentication

Si un consommateur d'assertions doit s'appuyer sur l'authentification d'un principal par une autorité d'authentification, le consommateur d'assertions peut avoir besoin d'informations additionnelles à l'assertion elle-même afin de s'assurer du niveau de confiance qu'il peut mettre dans cette assertion. La présente Recommandation définit un schéma XML pour la création de déclarations de contexte d'authentification – documents XML qui permettent à l'autorité d'authentification de fournir au consommateur d'assertions ces informations supplémentaires. De plus, la présente Recommandation définit un certain nombre de classes de contexte d'authentification; catégories dans lesquelles vont rentrer de nombreuses déclarations de contexte d'authentification, simplifiant par là leur interprétation.

SAML ne prescrit pas une seule technologie, protocole, ou politique pour les traitements par lesquels les autorités d'authentification produisent les identités aux principaux et par lesquels ces principaux s'authentifient ensuite auprès des autorités d'authentification. Différentes autorités d'authentification choisiront des technologies différentes, suivront des processus différents, et seront liées par des obligations légales différentes par rapport à la façon dont elles authentifient les principaux.

Les choix qu'une autorité d'authentification fait ici seront conduits dans une large mesure par les exigences des consommateurs d'assertions avec lesquels l'autorité d'authentification interagit. Ces exigences elles-mêmes seront déterminées par la nature du service (c'est-à-dire, au caractère sensible de l'information échangée, à la valeur financière associée, à la tolérance au risque des consommateurs d'assertions, etc.) que le consommateur d'assertions fournira au principal.

Par conséquent, pour tout ce qui n'est pas du service trivial, si le consommateur d'assertions doit avoir une confiance suffisante dans les assertions d'authentification qu'il reçoit d'une autorité d'authentification, il lui sera nécessaire de

savoir quelles technologies, quels protocoles et processus ont été utilisés ou suivis pour le mécanisme original d'authentification sur lequel l'assertion d'authentification se fonde. Armé de ces informations et confiant dans l'origine de l'assertion réelle, le consommateur d'assertions sera plus capable de prendre une décision justifiée concernant les services auxquels le sujet de l'assertion d'authentification devrait être admis à accéder.

Le contexte d'authentication se définit comme les informations, qui s'ajoutent à l'assertion d'authentification elle-même, que le consommateur d'assertions peut exiger avant de prendre une décision justifiée par rapport à une assertion d'authentification. Un tel contexte peut inclure, sans s'y limiter, la méthode réelle d'authentification utilisée.

12.2 Déclaration de contexte d'authentication

Si un consommateur d'assertions doit s'appuyer sur l'authentification d'une autre entité par une autorité d'authentification, le consommateur d'assertions peut exiger des informations additionnelles à l'authentification elle-même pour lui permettre de placer l'authentification dans un contexte de gestion du risque. Ces informations pourraient comporter:

- des mécanismes d'identification de l'utilisateur initial (par exemple, face à face, en ligne, secret partagé);
- des mécanismes pour minimiser la compromission des accréditifs (par exemple, fréquence de renouvellement des accréditifs, génération de clés côté client);
- des mécanismes pour mémoriser et protéger les accréditifs (par exemple, carte à mémoire, règles de mot de passe);
- le mécanisme ou la méthode d'authentification (par exemple, mot de passe).

Les variations et permutations des caractéristiques de la liste ci-dessus garantissent que toutes les assertions d'authentification n'auront pas le même degré de confiance qu'un consommateur d'assertions peut mettre en elles; en particulier, l'assertion d'authentification va être caractérisée par les valeurs de chacune de ces variables, et d'autres.

Une autorité d'authentification SAML peut délivrer à un consommateur d'assertions les informations additionnelles de contexte d'authentification sous la forme d'une déclaration de contexte d'authentification, un document XML inséré directement ou référencé au sein de l'assertion d'authentification, que l'autorité d'authentification fournit au consommateur d'assertions.

Les demandeurs SAML sont capables de demander qu'une authentification se conforme à un contexte d'authentification spécifié en identifiant ce contexte dans une demande d'authentification. Un demandeur peut aussi spécifier qu'une authentification doit être conduite dans un contexte d'authentification qui *excède* certaines valeurs déclarées (pour une définition donnée de "excède").

12.2.1 Modèle de données

Une déclaration de contexte d'authentification particulière définie dans la présente Recommandation va rassembler les caractéristiques des processus, procédures, et des mécanismes par lesquels l'autorité d'authentification a vérifié le sujet avant de produire une identité, protégé les secrets sur lesquels les authentifications suivantes sont fondées, et les mécanismes utilisés pour cette authentification. Ces caractéristiques sont catégorisées dans le schéma de contexte d'authentification comme suit:

- identification – Caractéristiques qui décrivent les processus et mécanismes que l'autorité d'authentification utilise pour créer initialement une association entre un sujet et l'identité (ou nom) par lequel le sujet sera connu.
- Protection technique – Caractéristiques qui décrivent comment le "secret" (dont la connaissance ou la possession permet au sujet de s'authentifier auprès de l'autorité d'authentification) est gardé en sécurité.
- Protection opérationnelle – Caractéristiques qui décrivent les contrôles de sécurité procédurale employés par l'autorité d'authentification (par exemple, audits de sécurité, enregistrement des archives).
- Méthode d'authentication – Caractéristiques qui définissent les mécanismes par lesquels le sujet de l'assertion produite s'authentifie auprès de l'autorité d'authentification (par exemple, un mot de passe ou une carte à mémoire).
- Accords constitutifs – Caractéristiques qui décrivent le cadre légal (c'est-à-dire les contraintes de responsabilité et les obligations contractuelles) sous-jacent à l'événement d'authentification et/ou son infrastructure d'authentification technique associée.

12.2.2 Extensibilité

La schéma de déclaration de contexte d'authentification a des points d'extensibilité bien définis par l'élément `<Extension>`. Les autorités d'authentification peuvent utiliser cet élément pour insérer des détails de contexte d'authentification supplémentaires pour les assertions SAML qu'elles produisent (en supposant que le consommateur

d'assertions sera capable de comprendre ces extensions). Ces éléments supplémentaires doivent être dans un espace de nom XML séparé de celui du schéma de base ou de classe de déclaration de contexte d'authentification qui s'applique à la déclaration elle-même.

12.2.3 Règles de traitement

Des règles de traitement supplémentaires de déclarations de contexte d'authentification sont spécifiées au § 8. Ces règles de traitement permettent aux développements de partager des interprétations communes de la force relative ou de la qualité des déclarations de contexte d'authentification particulières et ne peuvent pas être exprimées en termes absolus ou fournis comme règles que doivent suivre les implémentations.

12.2.4 Schéma

Le présent paragraphe n'est pas normatif.

La liste complète des schémas XML des types de contexte d'authentification et le schéma XML de contexte d'authentification lui-même, utilisée pour la validation des déclarations individuelles généralisées, est fournie à l'Appendice VI.

12.3 Classes de contexte d'authentication

Le nombre des permutations des différentes caractéristiques assure qu'il y a un nombre théoriquement infini de contextes d'authentification uniques. Cela implique, en théorie, que tout consommateur d'assertions particulier est réputé capable d'analyser une déclaration de contexte d'authentification arbitraire et, plus important, d'analyser la déclaration afin d'attester de la "qualité" de l'assertion d'authentification associée. Il n'est pas trivial de faire une telle attestation.

Une optimisation est heureusement possible. En pratique de nombreux contextes d'authentification vont entrer dans les catégories déterminées par les pratiques et les technologies de l'industrie. Par exemple, de nombreux contextes d'authentification de navigateurs de la toile de professionnel à grand public seront (partiellement) définis par l'authentification du principal auprès de l'autorité d'authentification à travers la présentation d'un mot de passe sur une session protégée par TLS. Dans le monde de l'entreprise, l'authentification fondée sur le certificat est courante. Bien sûr, le plein contexte d'authentification n'est pas limité aux spécificités des modalités de l'authentification du principal. Néanmoins, la méthode d'authentification est souvent la caractéristique la plus *visible* et comme telle, peut servir de classificateur utile pour une classe de contextes d'authentification en rapport.

Le concept est exprimé dans la présente Recommandation comme définition d'une série de *classes de contexte d'authentification.* Chaque classe définit un sous-ensemble approprié dans l'ensemble complet des contextes d'authentification. Les classes ont été choisies comme représentatives des pratiques et technologies courantes pour les techniques d'authentification, et elles fournissent aux producteurs et consommateurs d'assertions un raccourci convenable pour se référer aux questions de contexte d'authentification.

Par exemple, une autorité d'authentification peut inclure avec la déclaration de contexte d'authentification complète qu'il fournit à un consommateur d'assertions une assertion que le contexte d'authentification appartient aussi à une classe de contexte d'authentification. Pour certains consommateurs d'assertions, cette assertion est suffisement détaillée pour qu'il soit capable d'allouer un niveau de confiance approprié à l'assertion d'authentification associée. D'autres consommateurs d'assertions pourront préférer examiner la déclaration de contexte d'authentification complète elle-même. Vraisemblablement, la capacité à se référer à une classe de contexte d'authentification plutôt que d'être obligé de faire la liste de tous les détails d'une déclaration de contexte d'authentification spécifique simplifie la façon dont le consommateur d'assertions peut exprimer son désir et/ou ses exigences à une autorité d'authentification.

12.3.1 Avantages des classes de contexte d'authentification

L'introduction des couches supplémentaires de classes et la définition d'une liste initiale de classes représentatives et souples est destinée à:

- faciliter aux autorités d'authentification et aux consommateurs d'assertions la réalisation d'un accord sur les contextes d'authentification acceptables en leur donnant un cadre de discussion;
- faciliter aux consommateurs d'assertions l'indication de leurs préférences lorsqu'il demandent l'établissement de l'assertion d'authentification de la part d'une autorité d'authentification;
- simplifier pour les consommateurs d'assertions la charge du traitement des déclarations de contexte d'authentification en leur donnant l'option de se satisfaire de la classe associée;
- protéger les consommateurs d'assertions de l'impact de nouvelles techniques d'authentification;
- faciliter aux autorités d'authentification la publication de leurs capacités d'authentification, par exemple, par WSDL.

12.3.2 Règles de traitement

D'autres règles de traitement des classes de contexte d'authentification sont décrites au § 8. A la plupart des égards, ces règles de traitement aboutissent à ce que les développements partagent une interprétation commune de la force relative et de qualité des classes particulières de contexte d'authentification et elles ne peuvent être exprimées en termes absolus ou fournies comme des règles que doivent suivre les implémentations.

12.3.3 Extensibilité

Comme le fait le schéma central de déclaration de contexte d'authentification, les différents schémas de classe de contexte d'authentification permettent l'élément `<Extension>` dans certaines localisations de la structure arborescente. En général, lorsque l'élément `<Extension>` survient comme héritier de l'élément `<xs:choice>`, cette option a été retirée en créant la définition de schéma de classe appropriée comme restriction du type de base. Lorsque l'élément `<Extension>` survient comme héritier facultatif d'un élément `<xs:sequence>`, l'élément `<Extension>` peut rester en plus de tout élément obligatoire.

Par conséquent, les déclarations de contexte d'authentification peuvent inclure l'élément `<Extension>` (avec les éléments additionnels dans les différents espaces de nom) et se conformer encore aux schémas de classe de contexte d'authentification (s'il satisfont aux autres exigences du schéma, bien sûr).

Les schémas de classe de contexte d'authentification restreignent les définitions de type dans le schéma de base de contexte d'authentification. En tant que points d'extension, les schémas de classe de contexte d'authentification peuvent eux-mêmes être aussi restreints – leurs définitions de type servant de types de base dans certains autres schémas (éventuellement définis par des communautés qui souhaitent une classe de contexte d'authentification à définition plus stricte). Pour empêcher les incohérences logiques, de telles extensions de schéma peuvent seulement ajouter des contraintes aux définitions de type du schéma de classe. Pour mettre en application cette contrainte, les schémas de classe de contexte d'authentification sont définis avec l'attribut `finalDefault="extension"` sur l'élément `<schema>` pour empêcher ce type de déviation.

12.3.4 Schémas

La liste des classes de contexte d'authentification figure dans les paragraphes suivants. La liste des classes figure par ordre alphabétique; aucun autre classement n'est impliqué par l'ordre des classes. Les implémentations peuvent choisir quelles classes prendre en charge en fonction des lignes directrices de conformité données dans la présente Recommandation (voir § 13). Les classes sont identifiées de façon univoque par les URI avec la souche initiale suivante:

```
urn:oasis:names:tc:SAML:2.0:ac:classes
```

Les schémas de classe sont définis comme des restrictions des parties du schéma de base du contexte d'authentification "types". Les instances XML qui se valident par rapport à un schéma de classe de contexte d'authentification donné sont dites *conformes* à cette classe de contexte d'authentification.

Parce que le schéma de classe importe et redéfinit les éléments et types en nom d'espace de schéma de classe, une déclaration de contexte d'authentification conforme à une classe ne se valide pas simultanément par rapport au schéma de base de contexte d'authentification.

12.3.4.1 Protocole Internet

URI: `urn:oasis:names:tc:SAML:2.0:ac:classes:InternetProtocol`

Cet URI est aussi utilisé comme espace de nom cible dans le schéma de classe de contexte d'authentification de l'Annexe A.

La classe Protocole Internet s'applique lorsqu'un principal est authentifié par l'utilisation d'une adresse IP fournie.

```
<?xml version="1.0" encoding="UTF-8"?>

<xs:schema
  targetNamespace="urn:oasis:names:tc:SAML:2.0:ac:classes:InternetProtocol"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns="urn:oasis:names:tc:SAML:2.0:ac:classes:InternetProtocol"
  finalDefault="extension"
  blockDefault="substitution"
  version="2.0">

  <xs:redefine schemaLocation="saml-schema-authn-context-types-2.0.xsd">
```

```
    <xs:annotation>
      <xs:documentation>
        Class identifier: urn:oasis:names:tc:SAML:2.0:ac:classes:InternetProtocol
        Document identifier: saml-schema-authn-context-ip-2.0
        Location: http://docs.oasis-open.org/security/saml/v2.0/
        Revision history:
          V2.0 (March, 2005):
            New authentication context class schema for SAML V2.0.
      </xs:documentation>
    </xs:annotation>

    <xs:complexType name="AuthnContextDeclarationBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthnContextDeclarationBaseType">
          <xs:sequence>
            <xs:element ref="Identification" minOccurs="0"/>
            <xs:element ref="TechnicalProtection" minOccurs="0"/>
            <xs:element ref="OperationalProtection" minOccurs="0"/>
            <xs:element ref="AuthnMethod"/>
            <xs:element ref="GoverningAgreements" minOccurs="0"/>
            <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
          </xs:sequence>
          <xs:attribute name="ID" type="xs:ID" use="optional"/>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="AuthnMethodBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthnMethodBaseType">
          <xs:sequence>
            <xs:element ref="PrincipalAuthenticationMechanism" minOccurs="0"/>
            <xs:element ref="Authenticator"/>
            <xs:element ref="AuthenticatorTransportProtocol" minOccurs="0"/>
            <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="AuthenticatorBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthenticatorBaseType">
          <xs:sequence>
            <xs:element ref="IPAddress"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

  </xs:redefine>

</xs:schema>
```

12.3.4.2 InternetProtocolPassword

URI: `urn:oasis:names:tc:SAML:2.0:ac:classes:InternetProtocolPassword`

Noter que cet URI est aussi utilisé comme espace de nom cible dans le schéma de classe de contexte d'authentification de l'Annexe A.

La classe Mot de passe de protocole Internet s'applique lorsqu'un principal s'authentifie par l'utilisation d'une adresse IP fournie, en plus d'un nom d'usager/mot de passe.

```
<?xml version="1.0" encoding="UTF-8"?>

<xs:schema
  targetNamespace="urn:oasis:names:tc:SAML:2.0:ac:classes:InternetProtocolPassword"
  xmlns:ac="urn:oasis:names:tc:SAML:2.0:ac"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns="urn:oasis:names:tc:SAML:2.0:ac:classes:InternetProtocolPassword"
```

```
  finalDefault="extension"
  blockDefault="substitution"
  version="2.0">

  <xs:redefine schemaLocation="saml-schema-authn-context-types-2.0.xsd">

    <xs:annotation>
      <xs:documentation>
        Class identifier:
urn:oasis:names:tc:SAML:2.0:ac:classes:InternetProtocolPassword
        Document identifier: saml-schema-authn-context-ippword-2.0
        Location: http://docs.oasis-open.org/security/saml/v2.0/
        Revision history:
          V2.0 (March, 2005):
            New authentication context class schema for SAML V2.0.
      </xs:documentation>
    </xs:annotation>

    <xs:complexType name="AuthnContextDeclarationBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthnContextDeclarationBaseType">
          <xs:sequence>
            <xs:element ref="Identification" minOccurs="0"/>
            <xs:element ref="TechnicalProtection" minOccurs="0"/>
            <xs:element ref="OperationalProtection" minOccurs="0"/>
            <xs:element ref="AuthnMethod"/>
            <xs:element ref="GoverningAgreements" minOccurs="0"/>
            <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
          </xs:sequence>
          <xs:attribute name="ID" type="xs:ID" use="optional"/>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="AuthnMethodBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthnMethodBaseType">
          <xs:sequence>
            <xs:element ref="PrincipalAuthenticationMechanism" minOccurs="0"/>
            <xs:element ref="Authenticator"/>
            <xs:element ref="AuthenticatorTransportProtocol" minOccurs="0"/>
            <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="AuthenticatorBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthenticatorBaseType">
          <xs:sequence>
            <xs:element ref="Password"/>
            <xs:element ref="IPAddress"/>
            <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

  </xs:redefine>

</xs:schema>
```

12.3.4.3 Kerberos

URI: `urn:oasis:names:tc:SAML:2.0:ac:classes:Kerberos`

Cet URI est aussi utilisé comme espace de nom cible dans le schéma de classe de contexte d'authentification de l'Annexe A.

Cette classe s'applique lorsque le principal s'est authentifié en utilisant un mot de passe auprès d'une autorité d'authentification locale, afin d'acquérir un ticket Kerberos. Ce ticket Kerberos est ensuite utilisé pour l'authentification de réseau ultérieure.

NOTE 1 – Il est possible à l'autorité d'authentification d'indiquer (via cette classe de contexte) un type de données de pré-authentification qui a été utilisé par le centre de distribution de clés Kerberos (RFC 1510 de l'IETF) lors de l'authentification du principal. La méthode utilisée par l'autorité d'authentification pour obtenir ces informations sort du domaine d'application de la présente Recommandation, mais il est fortement recommandé qu'une méthode de confiance soit mise en œuvre pour passer le type de données de pré-authentification et tous détails de contexte en rapport avec Kerberos (par exemple, la durée de vie du ticket) à l'autorité d'authentification.

```
<?xml version="1.0" encoding="UTF-8"?>

<xs:schema
targetNamespace="urn:oasis:names:tc:SAML:2.0:ac:classes:Kerberos"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns="urn:oasis:names:tc:SAML:2.0:ac:classes:Kerberos"
  finalDefault="extension"
  blockDefault="substitution"
  version="2.0">

  <xs:redefine schemaLocation="saml-schema-authn-context-types-2.0.xsd">

    <xs:annotation>
      <xs:documentation>
        Class identifier: urn:oasis:names:tc:SAML:2.0:ac:classes:Kerberos
        Document identifier: saml-schema-authn-context-kerberos-2.0
        Location: http://docs.oasis-open.org/security/saml/v2.0/
        Revision history:
          V2.0 (March, 2005):
            New authentication context class schema for SAML V2.0.
      </xs:documentation>
    </xs:annotation>

    <xs:complexType name="AuthnContextDeclarationBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthnContextDeclarationBaseType">
          <xs:sequence>
            <xs:element ref="Identification" minOccurs="0"/>
            <xs:element ref="TechnicalProtection" minOccurs="0"/>
            <xs:element ref="OperationalProtection" minOccurs="0"/>
            <xs:element ref="AuthnMethod"/>
            <xs:element ref="GoverningAgreements" minOccurs="0"/>
            <xs:element ref="Extension" minOccurs="0"
maxOccurs="unbounded"/>
          </xs:sequence>
          <xs:attribute name="ID" type="xs:ID" use="optional"/>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="AuthnMethodBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthnMethodBaseType">
          <xs:sequence>
            <xs:element ref="PrincipalAuthenticationMechanism"/>
            <xs:element ref="Authenticator"/>
            <xs:element ref="AuthenticatorTransportProtocol"
minOccurs="0"/>
            <xs:element ref="Extension" minOccurs="0"
maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="PrincipalAuthenticationMechanismType">
      <xs:complexContent>
        <xs:restriction base="PrincipalAuthenticationMechanismType">
          <xs:sequence>
              <xs:element ref="RestrictedPassword"/>
```

```
          </xs:sequence>
          <xs:attribute name="preauth" type="xs:integer" use="optional"/>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="AuthenticatorBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthenticatorBaseType">
          <xs:sequence>
            <xs:element ref="SharedSecretChallengeResponse"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="SharedSecretChallengeResponseType">
      <xs:complexContent>
        <xs:restriction base="SharedSecretChallengeResponseType">
          <xs:attribute name="method" type="xs:anyURI"
fixed="urn:oasis:names:tc:SAML:2.0:ac:classes:Kerberos"/>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

  </xs:redefine>

</xs:schema>
```

Ci-après figure un exemple d'instance XML conforme à ce schéma de classe:

```
<AuthenticationContextDeclaration
  xmlns="urn:oasis:names:tc:SAML:2.0:ac:classes:Kerberos">

  <AuthnMethod>

    <PrincipalAuthenticationMechanism preauth="0">
      <RestrictedPassword>
        <Length min="4"/>
      </RestrictedPassword>
    </PrincipalAuthenticationMechanism>

    <Authenticator>
      <AuthenticatorSequence>
        <SharedSecretChallengeResponse
method="urn:oasis:names:tc:SAML:2.0:ac:classes:Kerberos"/>
      </AuthenticatorSequence>
    </Authenticator>

  </AuthnMethod>

</AuthenticationContextDeclaration>
```

NOTE 2 – L'utilisation de SSL est présentée à l'Appendice IV.

12.3.4.4 MobileOneFactorUnregistered

URI: `urn:oasis:names:tc:SAML:2.0:ac:classes:MobileOneFactorUnregistered`

Cet URI est aussi utilisé comme espace de nom cible dans le schéma de classe de contexte d'authentification de l'Annexe A.

Ne reflète aucune procédure d'enregistrement d'abonné mobile et une authentification de l'appareil mobile sans exiger d'interaction explicite avec l'utilisateur final. Cette classe de contexte n'authentifie que l'appareil et jamais l'usager; elle est utile lorsque des services autres que l'opérateur mobile veulent ajouter une authentification d'appareil sécurisée à leur processus d'authentification.

```
<?xml version="1.0" encoding="UTF-8"?>

<xs:schema

targetNamespace="urn:oasis:names:tc:SAML:2.0:ac:classes:MobileOneFactorUnre
gistered"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"

xmlns="urn:oasis:names:tc:SAML:2.0:ac:classes:MobileOneFactorUnregistered"
  finalDefault="extension"
  blockDefault="substitution"
  version="2.0">

  <xs:redefine schemaLocation="saml-schema-authn-context-types-2.0.xsd">

    <xs:annotation>
      <xs:documentation>
        Class identifier:
urn:oasis:names:tc:SAML:2.0:ac:classes:MobileOneFactorUnregistered
        Document identifier: saml-schema-authn-context-mobileonefactor-
unreg-2.0
        Location: http://docs.oasis-open.org/security/saml/v2.0/
        Revision history:
          V2.0 (March, 2005):
            New authentication context class schema for SAML V2.0.
      </xs:documentation>
    </xs:annotation>

    <xs:complexType name="AuthnContextDeclarationBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthnContextDeclarationBaseType">
          <xs:sequence>
            <xs:element ref="Identification" minOccurs="0"/>
            <xs:element ref="TechnicalProtection" minOccurs="0"/>
            <xs:element ref="OperationalProtection" minOccurs="0"/>
            <xs:element ref="AuthnMethod"/>
            <xs:element ref="GoverningAgreements" minOccurs="0"/>
            <xs:element ref="Extension" minOccurs="0"
maxOccurs="unbounded"/>
          </xs:sequence>
          <xs:attribute name="ID" type="xs:ID" use="optional"/>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="AuthnMethodBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthnMethodBaseType">
          <xs:sequence>
            <xs:element ref="PrincipalAuthenticationMechanism"
minOccurs="0"/>
            <xs:element ref="Authenticator"/>
            <xs:element ref="AuthenticatorTransportProtocol"
minOccurs="0"/>
            <xs:element ref="Extension" minOccurs="0"
maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="AuthenticatorBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthenticatorBaseType">
          <xs:sequence>
            <xs:choice>
              <xs:element ref="DigSig"/>
              <xs:element ref="ZeroKnowledge"/>
              <xs:element ref="SharedSecretChallengeResponse"/>
              <xs:element ref="SharedSecretDynamicPlaintext"/>
              <xs:element ref="AsymmetricDecryption"/>
```

```
              <xs:element ref="AsymmetricKeyAgreement"/>
            </xs:choice>
            <xs:element ref="Extension" minOccurs="0"
maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="AuthenticatorTransportProtocolType">
      <xs:complexContent>
        <xs:restriction base="AuthenticatorTransportProtocolType">
          <xs:sequence>
            <xs:choice>
              <xs:element ref="SSL"/>
              <xs:element ref="MobileNetworkNoEncryption"/>
              <xs:element ref="MobileNetworkRadioEncryption"/>
              <xs:element ref="MobileNetworkEndToEndEncryption"/>
              <xs:element ref="WTLS"/>
            </xs:choice>
            <xs:element ref="Extension" minOccurs="0"
maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="OperationalProtectionType">
      <xs:complexContent>
        <xs:restriction base="OperationalProtectionType">
          <xs:sequence>
            <xs:element ref="SecurityAudit"/>
            <xs:element ref="DeactivationCallCenter"/>
            <xs:element ref="Extension" minOccurs="0"
maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="TechnicalProtectionBaseType">
      <xs:complexContent>
        <xs:restriction base="TechnicalProtectionBaseType">
          <xs:sequence>
            <xs:choice>
              <xs:element ref="PrivateKeyProtection"/>
              <xs:element ref="SecretKeyProtection"/>
            </xs:choice>
            <xs:element ref="Extension" minOccurs="0"
maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="PrivateKeyProtectionType">
      <xs:complexContent>
        <xs:restriction base="PrivateKeyProtectionType">
          <xs:sequence>
            <xs:element ref="KeyStorage"/>
            <xs:element ref="Extension" minOccurs="0"
maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="SecretKeyProtectionType">
      <xs:complexContent>
        <xs:restriction base="SecretKeyProtectionType">
          <xs:sequence>
```

```
            <xs:element ref="KeyStorage"/>
            <xs:element ref="Extension" minOccurs="0"
maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="KeyStorageType">
      <xs:complexContent>
        <xs:restriction base="KeyStorageType">
          <xs:attribute name="medium" use="required">
            <xs:simpleType>
              <xs:restriction base="mediumType">
                <xs:enumeration value="MobileDevice"/>
                <xs:enumeration value="MobileAuthCard"/>
                <xs:enumeration value="smartcard"/>
              </xs:restriction>
            </xs:simpleType>
          </xs:attribute>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="SecurityAuditType">
      <xs:complexContent>
        <xs:restriction base="SecurityAuditType">
          <xs:sequence>
            <xs:element ref="SwitchAudit"/>
            <xs:element ref="Extension" minOccurs="0"
maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="IdentificationType">
      <xs:complexContent>
        <xs:restriction base="IdentificationType">
          <xs:sequence>
            <xs:element ref="GoverningAgreements"/>
            <xs:element ref="Extension" minOccurs="0"
maxOccurs="unbounded"/>
          </xs:sequence>
          <xs:attribute name="nym">
            <xs:simpleType>
              <xs:restriction base="nymType">
                <xs:enumeration value="anonymity"/>
                <xs:enumeration value="pseudonymity"/>
              </xs:restriction>
            </xs:simpleType>
          </xs:attribute>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

  </xs:redefine>

</xs:schema>
```

NOTE – L'utilisation de SSL est présentée à l'Appendice IV.

12.3.4.5 MobileTwoFactorUnregistered

URI: `urn:oasis:names:tc:SAML:2.0:ac:classes:MobileTwoFactorUnregistered`

Cet URI est aussi utilisé comme espace de nom cible dans le schéma de classe de contexte d'authentification de l'Annexe A.

Ne reflète aucune procédure d'enregistrement d'abonné mobile et une authentification fondée sur deux facteurs, tels qu'un appareil sécurisé et un PIN d'utilisateur. Cette classe de contexte est utile lorsqu'un service autre que l'opérateur

mobile veut lier son identifiant d'abonné à un service d'authentification à deux facteurs fourni par un mobile en capturant les données du téléphone mobile à l'inscription.

```
<?xml version="1.0" encoding="UTF-8"?>

<xs:schema
targetNamespace="urn:oasis:names:tc:SAML:2.0:ac:classes:MobileTwoFactorUnre
gistered"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"

xmlns="urn:oasis:names:tc:SAML:2.0:ac:classes:MobileTwoFactorUnregistered"
  finalDefault="extension"
  blockDefault="substitution"
  version="2.0">

  <xs:redefine schemaLocation="saml-schema-authn-context-types-2.0.xsd">

    <xs:annotation>
      <xs:documentation>
        Class identifier:
urn:oasis:names:tc:SAML:2.0:ac:classes:MobileTwoFactorUnregistered
        Document identifier: saml-schema-authn-context-mobiletwofactor-
unreg-2.0
        Location: http://docs.oasis-open.org/security/saml/v2.0/
        Revision history:
          V2.0 (March, 2005):
            New authentication context class schema for SAML V2.0.
      </xs:documentation>
    </xs:annotation>

    <xs:complexType name="AuthnContextDeclarationBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthnContextDeclarationBaseType">
          <xs:sequence>
            <xs:element ref="Identification" minOccurs="0"/>
            <xs:element ref="TechnicalProtection" minOccurs="0"/>
            <xs:element ref="OperationalProtection" minOccurs="0"/>
            <xs:element ref="AuthnMethod"/>
            <xs:element ref="GoverningAgreements" minOccurs="0"/>
            <xs:element ref="Extension" minOccurs="0"
maxOccurs="unbounded"/>
          </xs:sequence>
          <xs:attribute name="ID" type="xs:ID" use="optional"/>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="AuthnMethodBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthnMethodBaseType">
          <xs:sequence>
            <xs:element ref="PrincipalAuthenticationMechanism"
minOccurs="0"/>
            <xs:element ref="Authenticator"/>
            <xs:element ref="AuthenticatorTransportProtocol"
minOccurs="0"/>
            <xs:element ref="Extension" minOccurs="0"
maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="AuthenticatorBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthenticatorBaseType">
          <xs:sequence>
            <xs:choice>
              <xs:element ref="DigSig"/>
              <xs:element ref="ZeroKnowledge"/>
              <xs:element ref="SharedSecretChallengeResponse"/>
```

```
              <xs:element ref="SharedSecretDynamicPlaintext"/>
              <xs:element ref="AsymmetricDecryption"/>
              <xs:element ref="AsymmetricKeyAgreement"/>
              <xs:element ref="ComplexAuthenticator"/>
            </xs:choice>
            <xs:element ref="Extension" minOccurs="0"
maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="ComplexAuthenticatorType">
      <xs:complexContent>
        <xs:restriction base="ComplexAuthenticatorType">
          <xs:sequence>
            <xs:choice>
              <xs:element ref="SharedSecretChallengeResponse"/>
              <xs:element ref="SharedSecretDynamicPlaintext"/>
            </xs:choice>
            <xs:element ref="Password"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="AuthenticatorTransportProtocolType">
      <xs:complexContent>
        <xs:restriction base="AuthenticatorTransportProtocolType">
          <xs:sequence>
            <xs:choice>
              <xs:element ref="SSL"/>
              <xs:element ref="MobileNetworkNoEncryption"/>
              <xs:element ref="MobileNetworkRadioEncryption"/>
              <xs:element ref="MobileNetworkEndToEndEncryption"/>
              <xs:element ref="WTLS"/>
            </xs:choice>
            <xs:element ref="Extension" minOccurs="0"
maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="OperationalProtectionType">
      <xs:complexContent>
        <xs:restriction base="OperationalProtectionType">
          <xs:sequence>
            <xs:element ref="SecurityAudit"/>
            <xs:element ref="DeactivationCallCenter"/>
            <xs:element ref="Extension" minOccurs="0"
maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="TechnicalProtectionBaseType">
      <xs:complexContent>
        <xs:restriction base="TechnicalProtectionBaseType">
          <xs:sequence>
            <xs:choice>
              <xs:element ref="PrivateKeyProtection"/>
              <xs:element ref="SecretKeyProtection"/>
            </xs:choice>
            <xs:element ref="Extension" minOccurs="0"
maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>
```

```
    <xs:complexType name="PrivateKeyProtectionType">
      <xs:complexContent>
        <xs:restriction base="PrivateKeyProtectionType">
          <xs:sequence>
            <xs:element ref="KeyActivation"/>
            <xs:element ref="KeyStorage"/>
            <xs:element ref="Extension" minOccurs="0"
maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="SecretKeyProtectionType">
      <xs:complexContent>
        <xs:restriction base="SecretKeyProtectionType">
          <xs:sequence>
            <xs:element ref="KeyActivation"/>
            <xs:element ref="KeyStorage"/>
            <xs:element ref="Extension" minOccurs="0"
maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="KeyStorageType">
      <xs:complexContent>
        <xs:restriction base="KeyStorageType">
          <xs:attribute name="medium" use="required">
            <xs:simpleType>
              <xs:restriction base="mediumType">
                <xs:enumeration value="MobileDevice"/>
                <xs:enumeration value="MobileAuthCard"/>
                <xs:enumeration value="smartcard"/>
              </xs:restriction>
            </xs:simpleType>
          </xs:attribute>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="SecurityAuditType">
      <xs:complexContent>
        <xs:restriction base="SecurityAuditType">
          <xs:sequence>
            <xs:element ref="SwitchAudit"/>
            <xs:element ref="Extension" minOccurs="0"
maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="IdentificationType">
      <xs:complexContent>
        <xs:restriction base="IdentificationType">
          <xs:sequence>
            <xs:element ref="GoverningAgreements"/>
            <xs:element ref="Extension" minOccurs="0"
maxOccurs="unbounded"/>
          </xs:sequence>
          <xs:attribute name="nym">
            <xs:simpleType>
              <xs:restriction base="nymType">
                <xs:enumeration value="anonymity"/>
                <xs:enumeration value="pseudonymity"/>
              </xs:restriction>
            </xs:simpleType>
          </xs:attribute>
```

```
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

  </xs:redefine>

</xs:schema>
```

12.3.4.6 MobileOneFactorContract

URI: `urn:oasis:names:tc:SAML:2.0:ac:classes:MobileOneFactorContract`

Cet URI est aussi utilisé comme espace de nom cible dans le schéma de classe de contexte d'authentification de l'Annexe A.

Reflète les procédures d'enregistrement d'abonné à un contrat de mobile et un seul facteur d'authentification. Par exemple, un appareil de signature numérique avec une mémoire inviolable pour le stockage des clés, comme le MSISDN mobile, mais sans PIN ou éléments de biométrie exigés pour une authentification d'utilisateur en temps réel.

```
<?xml version="1.0" encoding="UTF-8"?>

<xs:schema

targetNamespace="urn:oasis:names:tc:SAML:2.0:ac:classes:MobileOneFactorContract
"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns="urn:oasis:names:tc:SAML:2.0:ac:classes:MobileOneFactorContract"
  finalDefault="extension"
  blockDefault="substitution"
  version="2.0">

  <xs:redefine schemaLocation="saml-schema-authn-context-types-2.0.xsd">

    <xs:annotation>
      <xs:documentation>
        Class identifier:
urn:oasis:names:tc:SAML:2.0:ac:classes:MobileOneFactorContract
        Document identifier: saml-schema-authn-context-mobileonefactor-reg-2.0
        Location: http://docs.oasis-open.org/security/saml/v2.0/
        Revision history:
          V2.0 (March, 2005):
            New authentication context class schema for SAML V2.0.
      </xs:documentation>
    </xs:annotation>

    <xs:complexType name="AuthnContextDeclarationBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthnContextDeclarationBaseType">
          <xs:sequence>
            <xs:element ref="Identification" minOccurs="0"/>
            <xs:element ref="TechnicalProtection" minOccurs="0"/>
            <xs:element ref="OperationalProtection" minOccurs="0"/>
            <xs:element ref="AuthnMethod"/>
            <xs:element ref="GoverningAgreements" minOccurs="0"/>
            <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
          </xs:sequence>
          <xs:attribute name="ID" type="xs:ID" use="optional"/>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="AuthnMethodBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthnMethodBaseType">
          <xs:sequence>
            <xs:element ref="PrincipalAuthenticationMechanism" minOccurs="0"/>
            <xs:element ref="Authenticator"/>
            <xs:element ref="AuthenticatorTransportProtocol" minOccurs="0"/>
            <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:restriction>
```

```
  </xs:complexContent>
</xs:complexType>

<xs:complexType name="AuthenticatorBaseType">
  <xs:complexContent>
    <xs:restriction base="AuthenticatorBaseType">
      <xs:sequence>
        <xs:choice>
          <xs:element ref="DigSig"/>
          <xs:element ref="ZeroKnowledge"/>
          <xs:element ref="SharedSecretChallengeResponse"/>
          <xs:element ref="SharedSecretDynamicPlaintext"/>
          <xs:element ref="AsymmetricDecryption"/>
          <xs:element ref="AsymmetricKeyAgreement"/>
        </xs:choice>
        <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
      </xs:sequence>
    </xs:restriction>
  </xs:complexContent>
</xs:complexType>

<xs:complexType name="AuthenticatorTransportProtocolType">
  <xs:complexContent>
    <xs:restriction base="AuthenticatorTransportProtocolType">
      <xs:sequence>
        <xs:choice>
          <xs:element ref="SSL"/>
          <xs:element ref="MobileNetworkNoEncryption"/>
          <xs:element ref="MobileNetworkRadioEncryption"/>
          <xs:element ref="MobileNetworkEndToEndEncryption"/>
          <xs:element ref="WTLS"/>
        </xs:choice>
        <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
      </xs:sequence>
    </xs:restriction>
  </xs:complexContent>
</xs:complexType>

<xs:complexType name="OperationalProtectionType">
  <xs:complexContent>
    <xs:restriction base="OperationalProtectionType">
      <xs:sequence>
        <xs:element ref="SecurityAudit"/>
        <xs:element ref="DeactivationCallCenter"/>
        <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
      </xs:sequence>
    </xs:restriction>
  </xs:complexContent>
</xs:complexType>

<xs:complexType name="TechnicalProtectionBaseType">
  <xs:complexContent>
    <xs:restriction base="TechnicalProtectionBaseType">
      <xs:sequence>
        <xs:choice>
          <xs:element ref="PrivateKeyProtection"/>
          <xs:element ref="SecretKeyProtection"/>
        </xs:choice>
        <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
      </xs:sequence>
    </xs:restriction>
  </xs:complexContent>
</xs:complexType>

<xs:complexType name="PrivateKeyProtectionType">
  <xs:complexContent>
    <xs:restriction base="PrivateKeyProtectionType">
      <xs:sequence>
        <xs:element ref="KeyStorage"/>
        <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
      </xs:sequence>
```

```
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="SecretKeyProtectionType">
      <xs:complexContent>
        <xs:restriction base="SecretKeyProtectionType">
          <xs:sequence>
            <xs:element ref="KeyStorage"/>
            <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="KeyStorageType">
      <xs:complexContent>
        <xs:restriction base="KeyStorageType">
          <xs:attribute name="medium" use="required">
            <xs:simpleType>
              <xs:restriction base="mediumType">
                <xs:enumeration value="smartcard"/>
                <xs:enumeration value="MobileDevice"/>
                <xs:enumeration value="MobileAuthCard"/>
              </xs:restriction>
            </xs:simpleType>
          </xs:attribute>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="SecurityAuditType">
      <xs:complexContent>
        <xs:restriction base="SecurityAuditType">
          <xs:sequence>
            <xs:element ref="SwitchAudit"/>
            <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="IdentificationType">
      <xs:complexContent>
        <xs:restriction base="IdentificationType">
          <xs:sequence>
            <xs:element ref="PhysicalVerification"/>
            <xs:element ref="WrittenConsent"/>
            <xs:element ref="GoverningAgreements"/>
            <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
          </xs:sequence>
          <xs:attribute name="nym">
            <xs:simpleType>
              <xs:restriction base="nymType">
                <xs:enumeration value="anonymity"/>
                <xs:enumeration value="verinymity"/>
                <xs:enumeration value="pseudonymity"/>
              </xs:restriction>
            </xs:simpleType>
          </xs:attribute>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

  </xs:redefine>

</xs:schema>
```

12.3.4.7 MobileTwoFactorContract

URI: `urn:oasis:names:tc:SAML:2.0:ac:classes:MobileTwoFactorContract`

Cet URI est aussi utilisé comme espace de nom cible dans le schéma de classe de contexte d'authentification de l'Annexe A.

Reflète les procédures d'enregistrement d'abonné à un contrat de mobile et une authentification fondée sur deux facteurs. Par exemple, un appareil de signature numérique avec une mémoire inviolable pour le stockage des clés, comme le SIM du GSM, qui exige une preuve explicite de l'identité et des intentions de l'utilisateur, tel qu'un PIN ou des éléments de biométrie.

```
<?xml version="1.0" encoding="UTF-8"?>

<xs:schema
targetNamespace="urn:oasis:names:tc:SAML:2.0:ac:classes:MobileTwoFactorCont
ract"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns="urn:oasis:names:tc:SAML:2.0:ac:classes:MobileTwoFactorContract"
  finalDefault="extension"
  blockDefault="substitution"
  version="2.0">

  <xs:redefine schemaLocation="saml-schema-authn-context-types-2.0.xsd">

    <xs:annotation>
      <xs:documentation>
        Class identifier:
urn:oasis:names:tc:SAML:2.0:ac:classes:MobileTwoFactorContract
        Document identifier: saml-schema-authn-context-mobiletwofactor-reg-
2.0
        Location: http://docs.oasis-open.org/security/saml/v2.0/
        Revision history:
          V2.0 (March, 2005):
            New authentication context class schema for SAML V2.0.
      </xs:documentation>
    </xs:annotation>

    <xs:complexType name="AuthnContextDeclarationBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthnContextDeclarationBaseType">
          <xs:sequence>
            <xs:element ref="Identification" minOccurs="0"/>
            <xs:element ref="TechnicalProtection" minOccurs="0"/>
            <xs:element ref="OperationalProtection" minOccurs="0"/>
            <xs:element ref="AuthnMethod"/>
            <xs:element ref="GoverningAgreements" minOccurs="0"/>
            <xs:element ref="Extension" minOccurs="0"
maxOccurs="unbounded"/>
          </xs:sequence>
          <xs:attribute name="ID" type="xs:ID" use="optional"/>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="AuthnMethodBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthnMethodBaseType">
          <xs:sequence>
            <xs:element ref="PrincipalAuthenticationMechanism"
minOccurs="0"/>
            <xs:element ref="Authenticator"/>
            <xs:element ref="AuthenticatorTransportProtocol"
minOccurs="0"/>
            <xs:element ref="Extension" minOccurs="0"
maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>
```

```
    <xs:complexType name="AuthenticatorBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthenticatorBaseType">
          <xs:sequence>
            <xs:choice>
              <xs:element ref="DigSig"/>
              <xs:element ref="ZeroKnowledge"/>
              <xs:element ref="SharedSecretChallengeResponse"/>
              <xs:element ref="SharedSecretDynamicPlaintext"/>
              <xs:element ref="AsymmetricDecryption"/>
              <xs:element ref="AsymmetricKeyAgreement"/>
              <xs:element ref="ComplexAuthenticator"/>
            </xs:choice>
            <xs:element ref="Extension" minOccurs="0"
maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="ComplexAuthenticatorType">
      <xs:complexContent>
        <xs:restriction base="ComplexAuthenticatorType">
          <xs:sequence>
            <xs:choice>
              <xs:element ref="SharedSecretChallengeResponse"/>
              <xs:element ref="SharedSecretDynamicPlaintext"/>
            </xs:choice>
            <xs:element ref="Password"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="AuthenticatorTransportProtocolType">
      <xs:complexContent>
        <xs:restriction base="AuthenticatorTransportProtocolType">
          <xs:sequence>
            <xs:choice>
              <xs:element ref="SSL"/>
              <xs:element ref="MobileNetworkNoEncryption"/>
              <xs:element ref="MobileNetworkRadioEncryption"/>
              <xs:element ref="MobileNetworkEndToEndEncryption"/>
              <xs:element ref="WTLS"/>
            </xs:choice>
            <xs:element ref="Extension" minOccurs="0"
maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="OperationalProtectionType">
      <xs:complexContent>
        <xs:restriction base="OperationalProtectionType">
          <xs:sequence>
            <xs:element ref="SecurityAudit"/>
            <xs:element ref="DeactivationCallCenter"/>
            <xs:element ref="Extension" minOccurs="0"
maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="TechnicalProtectionBaseType">
      <xs:complexContent>
        <xs:restriction base="TechnicalProtectionBaseType">
          <xs:sequence>
            <xs:choice>
              <xs:element ref="PrivateKeyProtection"/>
```

```
              <xs:element ref="SecretKeyProtection"/>
            </xs:choice>
            <xs:element ref="Extension" minOccurs="0"
maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="PrivateKeyProtectionType">
      <xs:complexContent>
        <xs:restriction base="PrivateKeyProtectionType">
          <xs:sequence>
            <xs:element ref="KeyActivation"/>
            <xs:element ref="KeyStorage"/>
            <xs:element ref="Extension" minOccurs="0"
maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="SecretKeyProtectionType">
      <xs:complexContent>
        <xs:restriction base="SecretKeyProtectionType">
          <xs:sequence>
            <xs:element ref="KeyActivation"/>
            <xs:element ref="KeyStorage"/>
            <xs:element ref="Extension" minOccurs="0"
maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="KeyStorageType">
      <xs:complexContent>
        <xs:restriction base="KeyStorageType">
          <xs:attribute name="medium" use="required">
            <xs:simpleType>
              <xs:restriction base="mediumType">
                <xs:enumeration value="MobileDevice"/>
                <xs:enumeration value="MobileAuthCard"/>
                <xs:enumeration value="smartcard"/>
              </xs:restriction>
            </xs:simpleType>
          </xs:attribute>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="SecurityAuditType">
      <xs:complexContent>
        <xs:restriction base="SecurityAuditType">
          <xs:sequence>
            <xs:element ref="SwitchAudit"/>
            <xs:element ref="Extension" minOccurs="0"
maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="IdentificationType">
      <xs:complexContent>
        <xs:restriction base="IdentificationType">
          <xs:sequence>
            <xs:element ref="PhysicalVerification"/>
            <xs:element ref="WrittenConsent"/>
            <xs:element ref="GoverningAgreements"/>
```

```
            <xs:element ref="Extension" minOccurs="0"
maxOccurs="unbounded"/>
          </xs:sequence>
          <xs:attribute name="nym">
            <xs:simpleType>
              <xs:restriction base="nymType">
                <xs:enumeration value="anonymity"/>
                <xs:enumeration value="verinymity"/>
                <xs:enumeration value="pseudonymity"/>
              </xs:restriction>
            </xs:simpleType>
          </xs:attribute>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

  </xs:redefine>

</xs:schema>
```

NOTE – L'utilisation de SSL est présentée à l'Appendice IV.

12.3.4.8 Password

URI: `urn:oasis:names:tc:SAML:2.0:ac:classes:Password`

Cet URI est aussi utilisé comme espace de nom cible dans le schéma de classe de contexte d'authentification de l'Annexe A.

La classe Password s'applique lorsqu'un principal s'authentifie auprès d'une autorité d'authentification par la présentation d'un mot de passe sur une session HTTP non protégée.

```
<?xml version="1.0" encoding="UTF-8"?>

<xs:schema
targetNamespace="urn:oasis:names:tc:SAML:2.0:ac:classes:Password"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns="urn:oasis:names:tc:SAML:2.0:ac:classes:Password"
  finalDefault="extension"
  blockDefault="substitution"
  version="2.0">

  <xs:redefine schemaLocation="saml-schema-authn-context-types-2.0.xsd">

    <xs:annotation>
      <xs:documentation>
        Class identifier: urn:oasis:names:tc:SAML:2.0:ac:classes:Password
        Document identifier: saml-schema-authn-context-pword-2.0
        Location: http://docs.oasis-open.org/security/saml/v2.0/
        Revision history:
          V2.0 (March, 2005):
            New authentication context class schema for SAML V2.0.
      </xs:documentation>
    </xs:annotation>

    <xs:complexType name="AuthnContextDeclarationBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthnContextDeclarationBaseType">
          <xs:sequence>
            <xs:element ref="Identification" minOccurs="0"/>
            <xs:element ref="TechnicalProtection" minOccurs="0"/>
            <xs:element ref="OperationalProtection" minOccurs="0"/>
            <xs:element ref="AuthnMethod"/>
            <xs:element ref="GoverningAgreements" minOccurs="0"/>
            <xs:element ref="Extension" minOccurs="0"
maxOccurs="unbounded"/>
          </xs:sequence>
          <xs:attribute name="ID" type="xs:ID" use="optional"/>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>
```

```
    <xs:complexType name="AuthnMethodBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthnMethodBaseType">
          <xs:sequence>
            <xs:element ref="PrincipalAuthenticationMechanism"
minOccurs="0"/>
            <xs:element ref="Authenticator"/>
            <xs:element ref="AuthenticatorTransportProtocol"
minOccurs="0"/>
            <xs:element ref="Extension" minOccurs="0"
maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="AuthenticatorBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthenticatorBaseType">
          <xs:sequence>
            <xs:element ref="RestrictedPassword"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

  </xs:redefine>

</xs:schema>
```

Ci-après figure un exemple d'instance XML qui se conforme au schéma de classe de contexte:

```
<AuthenticationContextDeclaration
  xmlns="urn:oasis:names:tc:SAML:2.0:ac:classes:Password">

  <AuthnMethod>
    <Authenticator>
      <AuthenticatorSequence>
        <RestrictedPassword>
          <Length min="4"/>
        </RestrictedPassword>
      </AuthenticatorSequence>
    </Authenticator>
  </AuthnMethod>

</AuthenticationContextDeclaration>
```

12.3.4.9 PasswordProtectedTransport

URI: `urn:oasis:names:tc:SAML:2.0:ac:classes:PasswordProtectedTransport`

Cet URI est aussi utilisé comme espace de nom cible dans le schéma de classe de contexte d'authentification de l'Annexe A.

La classe PasswordProtectedTransport s'applique lorsqu'un principal s'authentifie auprès d'une autorité d'authentification par la présentation d'un mot de passe sur une session protégée.

```
<?xml version="1.0" encoding="UTF-8"?>

<xs:schema
targetNamespace="urn:oasis:names:tc:SAML:2.0:ac:classes:PasswordProtectedTr
ansport"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns="urn:oasis:names:tc:SAML:2.0:ac:classes:PasswordProtectedTransport"
  finalDefault="extension"
  blockDefault="substitution"
  version="2.0">

  <xs:redefine schemaLocation="saml-schema-authn-context-types-2.0.xsd">
```

```
    <xs:annotation>
      <xs:documentation>
        Class identifier:
urn:oasis:names:tc:SAML:2.0:ac:classes:PasswordProtectedTransport
        Document identifier: saml-schema-authn-context-ppt-2.0
        Location: http://docs.oasis-open.org/security/saml/v2.0/
        Revision history:
          V2.0 (March, 2005):
            New authentication context class schema for SAML V2.0.
      </xs:documentation>
    </xs:annotation>

    <xs:complexType name="AuthnContextDeclarationBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthnContextDeclarationBaseType">
          <xs:sequence>
            <xs:element ref="Identification" minOccurs="0"/>
            <xs:element ref="TechnicalProtection" minOccurs="0"/>
            <xs:element ref="OperationalProtection" minOccurs="0"/>
            <xs:element ref="AuthnMethod"/>
            <xs:element ref="GoverningAgreements" minOccurs="0"/>
            <xs:element ref="Extension" minOccurs="0"
maxOccurs="unbounded"/>
          </xs:sequence>
          <xs:attribute name="ID" type="xs:ID" use="optional"/>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="AuthnMethodBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthnMethodBaseType">
          <xs:sequence>
            <xs:element ref="PrincipalAuthenticationMechanism"
minOccurs="0"/>
            <xs:element ref="Authenticator"/>
            <xs:element ref="AuthenticatorTransportProtocol"/>
            <xs:element ref="Extension" minOccurs="0"
maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="AuthenticatorBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthenticatorBaseType">
          <xs:sequence>
            <xs:element ref="RestrictedPassword"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="AuthenticatorTransportProtocolType">
      <xs:complexContent>
        <xs:restriction base="AuthenticatorTransportProtocolType">
          <xs:sequence>
            <xs:choice>
              <xs:element ref="SSL"/>
              <xs:element ref="MobileNetworkRadioEncryption"/>
              <xs:element ref="MobileNetworkEndToEndEncryption"/>
              <xs:element ref="WTLS"/>
              <xs:element ref="IPSec"/>
            </xs:choice>
            <xs:element ref="Extension" minOccurs="0"
maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>
```

```
  </xs:redefine>

</xs:schema>
```

NOTE – L'utilisation de SSL est présentée à l'Appendice IV.

12.3.4.10 PreviousSession

URI: `urn:oasis:names:tc:SAML:2.0:ac:classes:PreviousSession`

Cet URI est aussi utilisé comme espace de nom cible dans le schéma de classe de contexte d'authentification de l'Annexe A.

La classe PreviousSession s'applique lorsqu'un principal s'est authentifié auprès d'une autorité d'authentification à un moment passé en utilisant un contexte d'authentification pris en charge par cette autorité d'authentification. Par conséquent, un événement d'authentification ultérieur que l'autorité d'authentification va attester au consommateur d'assertions peut être significativement distant dans le temps de la demande actuelle d'accès à la ressource du principal.

Le contexte pour la session précédemment authentifiée n'est explicitement pas inclus dans cette classe de contexte parce que l'utilisateur ne s'est pas authentifié durant cette session, et donc le mécanisme utilisé par l'usager pour s'authentifier dans une session précédente ne devrait pas être utilisé au titre d'une décision sur la permission d'accès actuel à une ressource.

```
<?xml version="1.0" encoding="UTF-8"?>

<xs:schema
targetNamespace="urn:oasis:names:tc:SAML:2.0:ac:classes:PreviousSession"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns="urn:oasis:names:tc:SAML:2.0:ac:classes:PreviousSession"
  finalDefault="extension"
  blockDefault="substitution"
  version="2.0">

  <xs:redefine schemaLocation="saml-schema-authn-context-types-2.0.xsd">

    <xs:annotation>
      <xs:documentation>
        Class identifier:
urn:oasis:names:tc:SAML:2.0:ac:classes:PreviousSession
        Document identifier: saml-schema-authn-context-session-2.0
        Location: http://docs.oasis-open.org/security/saml/v2.0/
        Revision history:
          V2.0 (March, 2005):
            New authentication context class schema for SAML V2.0.
      </xs:documentation>
    </xs:annotation>

    <xs:complexType name="AuthnContextDeclarationBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthnContextDeclarationBaseType">
          <xs:sequence>
            <xs:element ref="Identification" minOccurs="0"/>
            <xs:element ref="TechnicalProtection" minOccurs="0"/>
            <xs:element ref="OperationalProtection" minOccurs="0"/>
            <xs:element ref="AuthnMethod"/>
            <xs:element ref="GoverningAgreements" minOccurs="0"/>
            <xs:element ref="Extension" minOccurs="0"
maxOccurs="unbounded"/>
          </xs:sequence>
          <xs:attribute name="ID" type="xs:ID" use="optional"/>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="AuthnMethodBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthnMethodBaseType">
          <xs:sequence>
```

```
            <xs:element ref="PrincipalAuthenticationMechanism"
minOccurs="0"/>
            <xs:element ref="Authenticator"/>
            <xs:element ref="AuthenticatorTransportProtocol"
minOccurs="0"/>
            <xs:element ref="Extension" minOccurs="0"
maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="AuthenticatorBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthenticatorBaseType">
          <xs:sequence>
            <xs:element ref="PreviousSession"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>
  </xs:redefine>
</xs:schema>
```

12.3.4.11 Clé publique – X.509

URI: `urn:oasis:names:tc:SAML:2.0:ac:classes:X509`

Cet URI est aussi utilisé comme espace de nom cible dans le schéma de classe de contexte d'authentification de l'Annexe A.

La classe de contexte X509 indique que le principal s'est authentifié au moyen d'une signature numérique où la clé a été validée au titre d'une infrastructure X.509 de clé publique.

```
<?xml version="1.0" encoding="UTF-8"?>

<xs:schema targetNamespace="urn:oasis:names:tc:SAML:2.0:ac:classes:X509"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns="urn:oasis:names:tc:SAML:2.0:ac:classes:X509"
  finalDefault="extension"
  blockDefault="substitution"
  version="2.0">

  <xs:redefine schemaLocation="saml-schema-authn-context-types-2.0.xsd">

    <xs:annotation>
      <xs:documentation>
        Class identifier: urn:oasis:names:tc:SAML:2.0:ac:classes:X509
        Document identifier: saml-schema-authn-context-x509-2.0
        Location: http://docs.oasis-open.org/security/saml/v2.0/
        Revision history:
          V2.0 (March, 2005):
            New authentication context class schema for SAML V2.0.
      </xs:documentation>
    </xs:annotation>

    <xs:complexType name="AuthnContextDeclarationBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthnContextDeclarationBaseType">
          <xs:sequence>
            <xs:element ref="Identification" minOccurs="0"/>
            <xs:element ref="TechnicalProtection" minOccurs="0"/>
            <xs:element ref="OperationalProtection" minOccurs="0"/>
            <xs:element ref="AuthnMethod"/>
            <xs:element ref="GoverningAgreements" minOccurs="0"/>
            <xs:element ref="Extension" minOccurs="0"
maxOccurs="unbounded"/>
          </xs:sequence>
          <xs:attribute name="ID" type="xs:ID" use="optional"/>
        </xs:restriction>
      </xs:complexContent>
```

```
    </xs:complexType>

    <xs:complexType name="AuthnMethodBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthnMethodBaseType">
          <xs:sequence>
            <xs:element ref="PrincipalAuthenticationMechanism"/>
            <xs:element ref="Authenticator"/>
            <xs:element ref="AuthenticatorTransportProtocol"
minOccurs="0"/>
            <xs:element ref="Extension" minOccurs="0"
maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="PrincipalAuthenticationMechanismType">
      <xs:complexContent>
        <xs:restriction base="PrincipalAuthenticationMechanismType">
          <xs:sequence>
            <xs:element ref="RestrictedPassword"/>
          </xs:sequence>
          <xs:attribute name="preauth" type="xs:integer" use="optional"/>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="AuthenticatorBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthenticatorBaseType">
          <xs:sequence>
            <xs:element ref="DigSig"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="PublicKeyType">
      <xs:complexContent>
        <xs:restriction base="PublicKeyType">
          <xs:attribute name="keyValidation" type="xs:anyURI"
fixed="urn:oasis:names:tc:SAML:2.0:ac:classes:X509"/>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

  </xs:redefine>

</xs:schema>
```

12.3.4.12 Clé publique – PGP

URI: `urn:oasis:names:tc:SAML:2.0:ac:classes:PGP`

Cet URI est aussi utilisé comme espace de nom cible dans le schéma de classe de contexte d'authentification de l'Annexe A.

La classe de contexte PGP indique que le principal s'est authentifié au moyen d'une signature numérique où la clé a été validée au titre d'une infrastructure PGP de clé publique.

```
<?xml version="1.0" encoding="UTF-8"?>

<xs:schema targetNamespace="urn:oasis:names:tc:SAML:2.0:ac:classes:PGP"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns="urn:oasis:names:tc:SAML:2.0:ac:classes:PGP"
  finalDefault="extension"
  blockDefault="substitution"
  version="2.0">

  <xs:redefine schemaLocation="saml-schema-authn-context-types-2.0.xsd">
```

```
    <xs:annotation>
      <xs:documentation>
        Class identifier: urn:oasis:names:tc:SAML:2.0:ac:classes:PGP
        Document identifier: saml-schema-authn-context-pgp-2.0
        Location: http://docs.oasis-open.org/security/saml/v2.0/
        Revision history:
          V2.0 (March, 2005):
            New authentication context class schema for SAML V2.0.
      </xs:documentation>
    </xs:annotation>

    <xs:complexType name="AuthnContextDeclarationBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthnContextDeclarationBaseType">
          <xs:sequence>
            <xs:element ref="Identification" minOccurs="0"/>
            <xs:element ref="TechnicalProtection" minOccurs="0"/>
            <xs:element ref="OperationalProtection" minOccurs="0"/>
            <xs:element ref="AuthnMethod"/>
            <xs:element ref="GoverningAgreements" minOccurs="0"/>
            <xs:element ref="Extension" minOccurs="0"
maxOccurs="unbounded"/>
          </xs:sequence>
          <xs:attribute name="ID" type="xs:ID" use="optional"/>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="AuthnMethodBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthnMethodBaseType">
          <xs:sequence>
            <xs:element ref="PrincipalAuthenticationMechanism"/>
            <xs:element ref="Authenticator"/>
            <xs:element ref="AuthenticatorTransportProtocol"
minOccurs="0"/>
            <xs:element ref="Extension" minOccurs="0"
maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="PrincipalAuthenticationMechanismType">
      <xs:complexContent>
        <xs:restriction base="PrincipalAuthenticationMechanismType">
          <xs:sequence>
            <xs:element ref="RestrictedPassword"/>
          </xs:sequence>
          <xs:attribute name="preauth" type="xs:integer" use="optional"/>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="AuthenticatorBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthenticatorBaseType">
          <xs:sequence>
            <xs:element ref="DigSig"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="PublicKeyType">
      <xs:complexContent>
        <xs:restriction base="PublicKeyType">
          <xs:attribute name="keyValidation"
fixed="urn:oasis:names:tc:SAML:2.0:ac:classes:PGP"/>
        </xs:restriction>
```

```
      </xs:complexContent>
    </xs:complexType>

  </xs:redefine>

</xs:schema>
```

12.3.4.13 Clé publique – SPKI

URI: `urn:oasis:names:tc:SAML:2.0:ac:classes:SPKI`

Cet URI est aussi utilisé comme espace de nom cible dans le schéma de classe de contexte d'authentification de l'Annexe A.

La classe de contexte SPKI indique que le principal s'est authentifié au moyen d'une signature numérique où la clé a été validée via une infrastructure SPKI.

```
<?xml version="1.0" encoding="UTF-8"?>

<xs:schema targetNamespace="urn:oasis:names:tc:SAML:2.0:ac:classes:SPKI"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns="urn:oasis:names:tc:SAML:2.0:ac:classes:SPKI"
  finalDefault="extension"
  blockDefault="substitution"
  version="2.0">

  <xs:redefine schemaLocation="saml-schema-authn-context-types-2.0.xsd">

    <xs:annotation>
      <xs:documentation>
        Class identifier: urn:oasis:names:tc:SAML:2.0:ac:classes:SPKI
        Document identifier: saml-schema-authn-context-spki-2.0
        Location: http://docs.oasis-open.org/security/saml/v2.0/
        Revision history:
          V2.0 (March, 2005):
            New authentication context class schema for SAML V2.0.
      </xs:documentation>
    </xs:annotation>

    <xs:complexType name="AuthnContextDeclarationBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthnContextDeclarationBaseType">
          <xs:sequence>
            <xs:element ref="Identification" minOccurs="0"/>
            <xs:element ref="TechnicalProtection" minOccurs="0"/>
            <xs:element ref="OperationalProtection" minOccurs="0"/>
            <xs:element ref="AuthnMethod"/>
            <xs:element ref="GoverningAgreements" minOccurs="0"/>
            <xs:element ref="Extension" minOccurs="0"
maxOccurs="unbounded"/>
          </xs:sequence>
          <xs:attribute name="ID" type="xs:ID" use="optional"/>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="AuthnMethodBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthnMethodBaseType">
          <xs:sequence>
            <xs:element ref="PrincipalAuthenticationMechanism"/>
            <xs:element ref="Authenticator"/>
            <xs:element ref="AuthenticatorTransportProtocol"
minOccurs="0"/>
            <xs:element ref="Extension" minOccurs="0"
maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>
```

```
    <xs:complexType name="PrincipalAuthenticationMechanismType">
      <xs:complexContent>
        <xs:restriction base="PrincipalAuthenticationMechanismType">
          <xs:sequence>
            <xs:element ref="RestrictedPassword"/>
          </xs:sequence>
          <xs:attribute name="preauth" type="xs:integer" use="optional"/>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="AuthenticatorBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthenticatorBaseType">
          <xs:sequence>
            <xs:element ref="DigSig"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="PublicKeyType">
      <xs:complexContent>
        <xs:restriction base="PublicKeyType">
          <xs:attribute name="keyValidation"
fixed="urn:oasis:names:tc:SAML:2.0:ac:classes:SPKI"/>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

  </xs:redefine>

</xs:schema>
```

12.3.4.14 Clé publique – Signature numérique XML

URI: `urn:oasis:names:tc:SAML:2.0:ac:classes:XMLDSig`

Cet URI est aussi utilisé comme espace de nom cible dans le schéma de classe de contexte d'authentification de l'Annexe A.

Cette classe de contexte indique que le principal s'est authentifié au moyen d'une signature numérique conformément aux règles de traitement spécifiées dans Signature XML du W3C.

```
<?xml version="1.0" encoding="UTF-8"?>

<xs:schema targetNamespace="urn:oasis:names:tc:SAML:2.0:ac:classes:XMLDSig"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns="urn:oasis:names:tc:SAML:2.0:ac:classes:XMLDSig"
  finalDefault="extension"
  blockDefault="substitution"
  version="2.0">

  <xs:redefine schemaLocation="saml-schema-authn-context-types-2.0.xsd">

    <xs:annotation>
      <xs:documentation>
        Class identifier: urn:oasis:names:tc:SAML:2.0:ac:classes:XMLDSig
        Document identifier: saml-schema-authn-context-xmldsig-2.0
        Location: http://docs.oasis-open.org/security/saml/v2.0/
        Revision history:
          V2.0 (March, 2005):
            New authentication context class schema for SAML V2.0.
      </xs:documentation>
    </xs:annotation>

    <xs:complexType name="AuthnContextDeclarationBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthnContextDeclarationBaseType">
          <xs:sequence>
            <xs:element ref="Identification" minOccurs="0"/>
```

```
            <xs:element ref="TechnicalProtection" minOccurs="0"/>
            <xs:element ref="OperationalProtection" minOccurs="0"/>
            <xs:element ref="AuthnMethod"/>
            <xs:element ref="GoverningAgreements" minOccurs="0"/>
            <xs:element ref="Extension" minOccurs="0"
maxOccurs="unbounded"/>
          </xs:sequence>
          <xs:attribute name="ID" type="xs:ID" use="optional"/>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="AuthnMethodBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthnMethodBaseType">
          <xs:sequence>
            <xs:element ref="PrincipalAuthenticationMechanism"/>
            <xs:element ref="Authenticator"/>
            <xs:element ref="AuthenticatorTransportProtocol"
minOccurs="0"/>
            <xs:element ref="Extension" minOccurs="0"
maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="PrincipalAuthenticationMechanismType">
      <xs:complexContent>
        <xs:restriction base="PrincipalAuthenticationMechanismType">
          <xs:sequence>
            <xs:element ref="RestrictedPassword"/>
          </xs:sequence>
          <xs:attribute name="preauth" type="xs:integer" use="optional"/>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="AuthenticatorBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthenticatorBaseType">
          <xs:sequence>
            <xs:element ref="DigSig"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="PublicKeyType">
      <xs:complexContent>
        <xs:restriction base="PublicKeyType">
          <xs:attribute name="keyValidation" type="xs:anyURI"
fixed="urn:ietf:rfc:3075"/>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

  </xs:redefine>

</xs:schema>
```

12.3.4.15 Smartcard

URI: `urn:oasis:names:tc:SAML:2.0:ac:classes:Smartcard`

Cet URI est aussi utilisé comme espace de nom cible dans le schéma de classe de contexte d'authentification de l'Annexe A.

La classe Smartcard est identifiée lorsqu'un principal s'authentifie auprès d'une autorité d'authentification en utilisant une carte à mémoire.

```
<?xml version="1.0" encoding="UTF-8"?>

<xs:schema
targetNamespace="urn:oasis:names:tc:SAML:2.0:ac:classes:Smartcard"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns="urn:oasis:names:tc:SAML:2.0:ac:classes:Smartcard"
  finalDefault="extension"
  blockDefault="substitution"
  version="2.0">

  <xs:redefine schemaLocation="saml-schema-authn-context-types-2.0.xsd">

    <xs:annotation>
      <xs:documentation>
        Class identifier: urn:oasis:names:tc:SAML:2.0:ac:classes:Smartcard
        Document identifier: saml-schema-authn-context-smartcard-2.0
        Location: http://docs.oasis-open.org/security/saml/v2.0/
        Revision history:
          V2.0 (March, 2005):
            New authentication context class schema for SAML V2.0.
      </xs:documentation>
    </xs:annotation>

    <xs:complexType name="AuthnContextDeclarationBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthnContextDeclarationBaseType">
          <xs:sequence>
            <xs:element ref="Identification" minOccurs="0"/>
            <xs:element ref="TechnicalProtection" minOccurs="0"/>
            <xs:element ref="OperationalProtection" minOccurs="0"/>
            <xs:element ref="AuthnMethod"/>
            <xs:element ref="GoverningAgreements" minOccurs="0"/>
            <xs:element ref="Extension" minOccurs="0"
maxOccurs="unbounded"/>
          </xs:sequence>
          <xs:attribute name="ID" type="xs:ID" use="optional"/>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="AuthnMethodBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthnMethodBaseType">
          <xs:sequence>
            <xs:element ref="PrincipalAuthenticationMechanism"/>
            <xs:element ref="Authenticator"/>
            <xs:element ref="AuthenticatorTransportProtocol"
minOccurs="0"/>
            <xs:element ref="Extension" minOccurs="0"
maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="PrincipalAuthenticationMechanismType">
      <xs:complexContent>
        <xs:restriction base="PrincipalAuthenticationMechanismType">
          <xs:sequence>
            <xs:element ref="Smartcard"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

  </xs:redefine>
</xs:schema>
```

12.3.4.16 SmartcardPKI

URI: `urn:oasis:names:tc:SAML:2.0:ac:classes:SmartcardPKI`

Cet URI est aussi utilisé comme espace de nom cible dans le schéma de classe de contexte d'authentification de l'Annexe A.

La classe SmartcardPKI s'applique lorsqu'un principal s'authentifie auprès d'une autorité d'authentification par un mécanisme d'authentification à deux facteurs utilisant une carte à mémoire avec une clé privée et un PIN incorporés.

```
<?xml version="1.0" encoding="UTF-8"?>

<xs:schema
targetNamespace="urn:oasis:names:tc:SAML:2.0:ac:classes:SmartcardPKI"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns="urn:oasis:names:tc:SAML:2.0:ac:classes:SmartcardPKI"
  finalDefault="extension"
  blockDefault="substitution"
  version="2.0">

  <xs:redefine schemaLocation="saml-schema-authn-context-types-2.0.xsd">

    <xs:annotation>
      <xs:documentation>
        Class identifier:
urn:oasis:names:tc:SAML:2.0:ac:classes:SmartcardPKI
        Document identifier: saml-schema-authn-context-smartcardpki-2.0
        Location: http://docs.oasis-open.org/security/saml/v2.0/
        Revision history:
          V2.0 (March, 2005):
            New authentication context class schema for SAML V2.0.
      </xs:documentation>
    </xs:annotation>

    <xs:complexType name="AuthnContextDeclarationBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthnContextDeclarationBaseType">
          <xs:sequence>
            <xs:element ref="Identification" minOccurs="0"/>
            <xs:element ref="TechnicalProtection"/>
            <xs:element ref="OperationalProtection" minOccurs="0"/>
            <xs:element ref="AuthnMethod"/>
            <xs:element ref="GoverningAgreements" minOccurs="0"/>
            <xs:element ref="Extension" minOccurs="0"
maxOccurs="unbounded"/>
          </xs:sequence>
          <xs:attribute name="ID" type="xs:ID" use="optional"/>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="AuthnMethodBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthnMethodBaseType">
          <xs:sequence>
            <xs:element ref="PrincipalAuthenticationMechanism"/>
            <xs:element ref="Authenticator"/>
            <xs:element ref="AuthenticatorTransportProtocol"
minOccurs="0"/>
            <xs:element ref="Extension" minOccurs="0"
maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="TechnicalProtectionBaseType">
      <xs:complexContent>
        <xs:restriction base="TechnicalProtectionBaseType">
          <xs:sequence>
            <xs:choice>
```

```
              <xs:element ref="PrivateKeyProtection"/>
            </xs:choice>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="PrincipalAuthenticationMechanismType">
      <xs:complexContent>
        <xs:restriction base="PrincipalAuthenticationMechanismType">
          <xs:sequence>
            <xs:element ref="Smartcard"/>
            <xs:element ref="ActivationPin"/>
            <xs:element ref="Extension" minOccurs="0"
maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="AuthenticatorBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthenticatorBaseType">
          <xs:sequence>
            <xs:choice>
              <xs:element ref="DigSig"/>
              <xs:element ref="AsymmetricDecryption"/>
              <xs:element ref="AsymmetricKeyAgreement"/>
            </xs:choice>
            <xs:element ref="Extension" minOccurs="0"
maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="PrivateKeyProtectionType">
      <xs:complexContent>
        <xs:restriction base="PrivateKeyProtectionType">
          <xs:sequence>
            <xs:element ref="KeyActivation"/>
            <xs:element ref="KeyStorage"/>
            <xs:element ref="Extension" minOccurs="0"
maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="KeyActivationType">
      <xs:complexContent>
        <xs:restriction base="KeyActivationType">
          <xs:sequence>
            <xs:element ref="ActivationPin"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="KeyStorageType">
      <xs:complexContent>
        <xs:restriction base="KeyStorageType">
          <xs:attribute name="medium" use="required">
            <xs:simpleType>
              <xs:restriction base="mediumType">
                <xs:enumeration value="smartcard"/>
              </xs:restriction>
            </xs:simpleType>
          </xs:attribute>
        </xs:restriction>
      </xs:complexContent>
```

```
    </xs:complexType>

  </xs:redefine>

</xs:schema>
```

12.3.4.17 SoftwarePKI

URI: `urn:oasis:names:tc:SAML:2.0:ac:classes:SoftwarePKI`

Cet URI est aussi utilisé comme espace de nom cible dans le schéma de classe de contexte d'authentification de l'Annexe A.

La classe SoftwarePKI s'applique lorsqu'un principal utilise un certificat X.509 mémorisé dans le logiciel pour s'authentifier auprès de l'autorité d'authentification.

```
<?xml version="1.0" encoding="UTF-8"?>

<xs:schema
targetNamespace="urn:oasis:names:tc:SAML:2.0:ac:classes:SoftwarePKI"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns="urn:oasis:names:tc:SAML:2.0:ac:classes:SoftwarePKI"
  finalDefault="extension"
  blockDefault="substitution"
  version="2.0">

  <xs:redefine schemaLocation="saml-schema-authn-context-types-2.0.xsd">

    <xs:annotation>
      <xs:documentation>
        Class identifier:
urn:oasis:names:tc:SAML:2.0:ac:classes:SoftwarePKI
        Document identifier: saml-schema-authn-context-softwarepki-2.0
        Location: http://docs.oasis-open.org/security/saml/v2.0/
        Revision history:
          V2.0 (March, 2005):
            New authentication context class schema for SAML V2.0.
      </xs:documentation>
    </xs:annotation>

    <xs:complexType name="AuthnContextDeclarationBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthnContextDeclarationBaseType">
          <xs:sequence>
            <xs:element ref="Identification" minOccurs="0"/>
            <xs:element ref="TechnicalProtection"/>
            <xs:element ref="OperationalProtection" minOccurs="0"/>
            <xs:element ref="AuthnMethod"/>
            <xs:element ref="GoverningAgreements" minOccurs="0"/>
            <xs:element ref="Extension" minOccurs="0"
maxOccurs="unbounded"/>
          </xs:sequence>
          <xs:attribute name="ID" type="xs:ID" use="optional"/>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="AuthnMethodBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthnMethodBaseType">
          <xs:sequence>
            <xs:element ref="PrincipalAuthenticationMechanism"/>
            <xs:element ref="Authenticator"/>
            <xs:element ref="AuthenticatorTransportProtocol"
minOccurs="0"/>
            <xs:element ref="Extension" minOccurs="0"
maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>
```

```
    <xs:complexType name="TechnicalProtectionBaseType">
      <xs:complexContent>
        <xs:restriction base="TechnicalProtectionBaseType">
          <xs:sequence>
            <xs:choice>
              <xs:element ref="PrivateKeyProtection"/>
            </xs:choice>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="PrincipalAuthenticationMechanismType">
      <xs:complexContent>
        <xs:restriction base="PrincipalAuthenticationMechanismType">
          <xs:sequence>
            <xs:element ref="ActivationPin"/>
            <xs:element ref="Extension" minOccurs="0"
maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="AuthenticatorBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthenticatorBaseType">
          <xs:sequence>
            <xs:choice>
              <xs:element ref="DigSig"/>
              <xs:element ref="AsymmetricDecryption"/>
              <xs:element ref="AsymmetricKeyAgreement"/>
            </xs:choice>
            <xs:element ref="Extension" minOccurs="0"
maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="PrivateKeyProtectionType">
      <xs:complexContent>
        <xs:restriction base="PrivateKeyProtectionType">
          <xs:sequence>
            <xs:element ref="KeyActivation"/>
            <xs:element ref="KeyStorage"/>
            <xs:element ref="Extension" minOccurs="0"
maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="KeyActivationType">
      <xs:complexContent>
        <xs:restriction base="KeyActivationType">
          <xs:sequence>
            <xs:element ref="ActivationPin"/>
            <xs:element ref="Extension" minOccurs="0"
maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="KeyStorageType">
      <xs:complexContent>
        <xs:restriction base="KeyStorageType">
          <xs:attribute name="medium" use="required">
            <xs:simpleType>
```

```
              <xs:restriction base="mediumType">
                <xs:enumeration value="memory"/>
              </xs:restriction>
            </xs:simpleType>
          </xs:attribute>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

  </xs:redefine>
</xs:schema>
```

12.3.4.18 Téléphonie

URI: `urn:oasis:names:tc:SAML:2.0:ac:classes:Telephony`

Cet URI est aussi utilisé comme espace de nom cible dans le schéma de classe de contexte d'authentification de l'Annexe A.

Cette classe est utilisée pour indiquer que le principal s'est authentifié via la fourniture d'un numéro téléphonique d'une ligne fixe, transporté via un protocole de téléphonie tel que l'ADSL.

```
<?xml version="1.0" encoding="UTF-8"?>

<xs:schema
targetNamespace="urn:oasis:names:tc:SAML:2.0:ac:classes:Telephony"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns="urn:oasis:names:tc:SAML:2.0:ac:classes:Telephony"
  finalDefault="extension"
  blockDefault="substitution"
  version="2.0">

  <xs:redefine schemaLocation="saml-schema-authn-context-types-2.0.xsd">

    <xs:annotation>
      <xs:documentation>
        Class identifier: urn:oasis:names:tc:SAML:2.0:ac:classes:Telephony
        Document identifier: saml-schema-authn-context-telephony-2.0
        Location: http://docs.oasis-open.org/security/saml/v2.0/
        Revision history:
          V2.0 (March, 2005):
            New authentication context class schema for SAML V2.0.
      </xs:documentation>
    </xs:annotation>

    <xs:complexType name="AuthnContextDeclarationBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthnContextDeclarationBaseType">
          <xs:sequence>
            <xs:element ref="Identification" minOccurs="0"/>
            <xs:element ref="TechnicalProtection" minOccurs="0"/>
            <xs:element ref="OperationalProtection" minOccurs="0"/>
            <xs:element ref="AuthnMethod"/>
            <xs:element ref="GoverningAgreements" minOccurs="0"/>
            <xs:element ref="Extension" minOccurs="0"
maxOccurs="unbounded"/>
          </xs:sequence>
          <xs:attribute name="ID" type="xs:ID" use="optional"/>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="AuthnMethodBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthnMethodBaseType">
          <xs:sequence>
            <xs:element ref="PrincipalAuthenticationMechanism"
minOccurs="0"/>
            <xs:element ref="Authenticator"/>
            <xs:element ref="AuthenticatorTransportProtocol"/>
```

```
            <xs:element ref="Extension" minOccurs="0"
maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="AuthenticatorBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthenticatorBaseType">
          <xs:sequence>
            <xs:element ref="SubscriberLineNumber"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="AuthenticatorTransportProtocolType">
      <xs:complexContent>
        <xs:restriction base="AuthenticatorTransportProtocolType">
          <xs:sequence>
            <xs:choice>
              <xs:element ref="PSTN"/>
              <xs:element ref="ISDN"/>
              <xs:element ref="ADSL"/>
            </xs:choice>
            <xs:element ref="Extension" minOccurs="0"
maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

  </xs:redefine>

</xs:schema>
```

12.3.4.19 Téléphonie (nomade)

URI: `urn:oasis:names:tc:SAML:2.0:ac:classes:NomadTelephony`

Cet URI est aussi utilisé comme espace de nom cible dans le schéma de classe de contexte d'authentification de l'Annexe A.

Indique que le principal est "en itinérance" (peut-être avec une carte téléphonique) et s'authentifie au moyen d'un numéro de ligne, d'un suffixe d'utilisateur, et d'un élément mot de passe.

```
<?xml version="1.0" encoding="UTF-8"?>

<xs:schema
targetNamespace="urn:oasis:names:tc:SAML:2.0:ac:classes:NomadTelephony"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns="urn:oasis:names:tc:SAML:2.0:ac:classes:NomadTelephony"
  finalDefault="extension"
  blockDefault="substitution"
  version="2.0">

  <xs:redefine schemaLocation="saml-schema-authn-context-types-2.0.xsd">

    <xs:annotation>
      <xs:documentation>
        Class identifier:
urn:oasis:names:tc:SAML:2.0:ac:classes:NomadTelephony
        Document identifier: saml-schema-authn-context-nomad-telephony-2.0
        Location: http://docs.oasis-open.org/security/saml/v2.0/
        Revision history:
          V2.0 (March, 2005):
            New authentication context class schema for SAML V2.0.
      </xs:documentation>
    </xs:annotation>
```

```
    <xs:complexType name="AuthnContextDeclarationBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthnContextDeclarationBaseType">
          <xs:sequence>
            <xs:element ref="Identification" minOccurs="0"/>
            <xs:element ref="TechnicalProtection" minOccurs="0"/>
            <xs:element ref="OperationalProtection" minOccurs="0"/>
            <xs:element ref="AuthnMethod"/>
            <xs:element ref="GoverningAgreements" minOccurs="0"/>
            <xs:element ref="Extension" minOccurs="0"
maxOccurs="unbounded"/>
          </xs:sequence>
          <xs:attribute name="ID" type="xs:ID" use="optional"/>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="AuthnMethodBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthnMethodBaseType">
          <xs:sequence>
            <xs:element ref="PrincipalAuthenticationMechanism"
minOccurs="0"/>
            <xs:element ref="Authenticator"/>
            <xs:element ref="AuthenticatorTransportProtocol"/>
            <xs:element ref="Extension" minOccurs="0"
maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="AuthenticatorBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthenticatorBaseType">
          <xs:sequence>
            <xs:element ref="Password"/>
            <xs:element ref="SubscriberLineNumber"/>
            <xs:element ref="UserSuffix"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="AuthenticatorTransportProtocolType">
      <xs:complexContent>
        <xs:restriction base="AuthenticatorTransportProtocolType">
          <xs:sequence>
            <xs:choice>
              <xs:element ref="PSTN"/>
              <xs:element ref="ISDN"/>
              <xs:element ref="ADSL"/>
            </xs:choice>
            <xs:element ref="Extension" minOccurs="0"
maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

  </xs:redefine>

</xs:schema>
```

12.3.4.20 Téléphonie (personnalisée)

URI: `urn:oasis:names:tc:SAML:2.0:ac:classes:PersonalTelephony`

Cet URI est aussi utilisé comme espace de nom cible dans le schéma de classe de contexte d'authentification de l'Annexe A.

Cette classe est utilisée pour indiquer que le principal s'est authentifié via la fourniture d'un numéro de téléphone de ligne fixe et d'un suffixe d'utilisateur, transportés via un protocole de téléphonie tel que l'ADSL.

```
<?xml version="1.0" encoding="UTF-8"?>

<xs:schema
targetNamespace="urn:oasis:names:tc:SAML:2.0:ac:classes:PersonalizedTelephony"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns="urn:oasis:names:tc:SAML:2.0:ac:classes:PersonalizedTelephony"
  finalDefault="extension"
  blockDefault="substitution"
  version="2.0">

  <xs:redefine schemaLocation="saml-schema-authn-context-types-2.0.xsd">

    <xs:annotation>
      <xs:documentation>
        Class identifier:
urn:oasis:names:tc:SAML:2.0:ac:classes:PersonalizedTelephony
        Document identifier: saml-schema-authn-context-personal-telephony-2.0
        Location: http://docs.oasis-open.org/security/saml/v2.0/
        Revision history:
          V2.0 (March, 2005):
            New authentication context class schema for SAML V2.0.
      </xs:documentation>
    </xs:annotation>

    <xs:complexType name="AuthnContextDeclarationBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthnContextDeclarationBaseType">
          <xs:sequence>
            <xs:element ref="Identification" minOccurs="0"/>
            <xs:element ref="TechnicalProtection" minOccurs="0"/>
            <xs:element ref="OperationalProtection" minOccurs="0"/>
            <xs:element ref="AuthnMethod"/>
            <xs:element ref="GoverningAgreements" minOccurs="0"/>
            <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
          </xs:sequence>
          <xs:attribute name="ID" type="xs:ID" use="optional"/>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="AuthnMethodBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthnMethodBaseType">
          <xs:sequence>
            <xs:element ref="PrincipalAuthenticationMechanism" minOccurs="0"/>
            <xs:element ref="Authenticator"/>
            <xs:element ref="AuthenticatorTransportProtocol"/>
            <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="AuthenticatorBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthenticatorBaseType">
          <xs:sequence>
            <xs:element ref="SubscriberLineNumber"/>
            <xs:element ref="UserSuffix"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="AuthenticatorTransportProtocolType">
      <xs:complexContent>
        <xs:restriction base="AuthenticatorTransportProtocolType">
          <xs:sequence>
```

```
            <xs:choice>
              <xs:element ref="PSTN"/>
              <xs:element ref="ISDN"/>
              <xs:element ref="ADSL"/>
            </xs:choice>
            <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

  </xs:redefine>

</xs:schema>
```

12.3.4.21 Téléphonie (authentifiée)

URI: `urn:oasis:names:tc:SAML:2.0:ac:classes:AuthenticatedTelephony`

Cet URI est aussi utilisé comme espace de nom cible dans le schéma de classe de contexte d'authentification de l'Annexe A.

Indique que le principal s'est authentifié au moyen d'un numéro de ligne, d'un suffixe d'usager, et d'un élément mot de passe.

```
<?xml version="1.0" encoding="UTF-8"?>

<xs:schema
targetNamespace="urn:oasis:names:tc:SAML:2.0:ac:classes:AuthenticatedTeleph
ony"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns="urn:oasis:names:tc:SAML:2.0:ac:classes:AuthenticatedTelephony"
  finalDefault="extension"
  blockDefault="substitution"
  version="2.0">

  <xs:redefine schemaLocation="saml-schema-authn-context-types-2.0.xsd">

    <xs:annotation>
      <xs:documentation>
        Class identifier:
urn:oasis:names:tc:SAML:2.0:ac:classes:AuthenticatedTelephony
        Document identifier: saml-schema-authn-context-auth-telephony-2.0
        Location: http://docs.oasis-open.org/security/saml/v2.0/
        Revision history:
          V2.0 (March, 2005):
            New authentication context class schema for SAML V2.0.
      </xs:documentation>
    </xs:annotation>

    <xs:complexType name="AuthnContextDeclarationBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthnContextDeclarationBaseType">
          <xs:sequence>
            <xs:element ref="Identification" minOccurs="0"/>
            <xs:element ref="TechnicalProtection" minOccurs="0"/>
            <xs:element ref="OperationalProtection" minOccurs="0"/>
            <xs:element ref="AuthnMethod"/>
            <xs:element ref="GoverningAgreements" minOccurs="0"/>
            <xs:element ref="Extension" minOccurs="0"
maxOccurs="unbounded"/>
          </xs:sequence>
          <xs:attribute name="ID" type="xs:ID" use="optional"/>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="AuthnMethodBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthnMethodBaseType">
          <xs:sequence>
```

```
            <xs:element ref="PrincipalAuthenticationMechanism"
minOccurs="0"/>
            <xs:element ref="Authenticator"/>
            <xs:element ref="AuthenticatorTransportProtocol"/>
            <xs:element ref="Extension" minOccurs="0"
maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>
    <xs:complexType name="AuthenticatorBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthenticatorBaseType">
          <xs:sequence>
            <xs:element ref="Password"/>
            <xs:element ref="SubscriberLineNumber"/>
            <xs:element ref="UserSuffix"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="AuthenticatorTransportProtocolType">
      <xs:complexContent>
        <xs:restriction base="AuthenticatorTransportProtocolType">
          <xs:sequence>
            <xs:choice>
              <xs:element ref="PSTN"/>
              <xs:element ref="ISDN"/>
              <xs:element ref="ADSL"/>
            </xs:choice>
            <xs:element ref="Extension" minOccurs="0"
maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>
  </xs:redefine>

</xs:schema>
```

12.3.4.22 Mot de passe distant sécurisé

URI: `urn:oasis:names:tc:SAML:2.0:ac:classes:SecureRemotePassword`

Cet URI est aussi utilisé comme espace de nom cible dans le schéma de classe de contexte d'authentification de l'Annexe A.

La classe SecureRemotePassword est applicable lorsque l'authentification a été effectuée au moyen d'un mot de passe distant sécurisé comme spécifié dans la RFC 2945 de l'IETF.

```
<?xml version="1.0" encoding="UTF-8"?>

<xs:schema
targetNamespace="urn:oasis:names:tc:SAML:2.0:ac:classes:SecureRemotePassword"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns="urn:oasis:names:tc:SAML:2.0:ac:classes:SecureRemotePassword"
  finalDefault="extension"
  blockDefault="substitution"
  version="2.0">

  <xs:redefine schemaLocation="saml-schema-authn-context-types-2.0.xsd">

    <xs:annotation>
      <xs:documentation>
        Class identifier: urn:oasis:names:tc:SAML:2.0:ac:classes:SecureRemotePassword
        Document identifier: saml-schema-authn-context-srp-2.0
        Location: http://docs.oasis-open.org/security/saml/v2.0/
        Revision history:
          V2.0 (March, 2005):
            New authentication context class schema for SAML V2.0.
```

```
      </xs:documentation>
    </xs:annotation>

    <xs:complexType name="AuthnContextDeclarationBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthnContextDeclarationBaseType">
          <xs:sequence>
            <xs:element ref="Identification" minOccurs="0"/>
            <xs:element ref="TechnicalProtection" minOccurs="0"/>
            <xs:element ref="OperationalProtection" minOccurs="0"/>
            <xs:element ref="AuthnMethod"/>
            <xs:element ref="GoverningAgreements" minOccurs="0"/>
            <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
          </xs:sequence>
          <xs:attribute name="ID" type="xs:ID" use="optional"/>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="AuthnMethodBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthnMethodBaseType">
          <xs:sequence>
            <xs:element ref="PrincipalAuthenticationMechanism"/>
            <xs:element ref="Authenticator"/>
            <xs:element ref="AuthenticatorTransportProtocol" minOccurs="0"/>
            <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="PrincipalAuthenticationMechanismType">
      <xs:complexContent>
        <xs:restriction base="PrincipalAuthenticationMechanismType">
          <xs:sequence>
            <xs:element ref="RestrictedPassword"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="AuthenticatorBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthenticatorBaseType">
          <xs:sequence>
            <xs:element ref="SharedSecretChallengeResponse"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="SharedSecretChallengeResponseType">
      <xs:complexContent>
        <xs:restriction base="SharedSecretChallengeResponseType">
          <xs:attribute name="method" type="xs:anyURI" fixed="urn:ietf:rfc:2945"/>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

  </xs:redefine>

</xs:schema>
```

12.3.4.23 Authentification client fondée sur un certificat TLS

URI: `urn:oasis:names:tc:SAML:2.0:ac:classes:TLSClient`

Cet URI est aussi utilisé comme espace de nom cible dans le schéma de classe de contexte d'authentification de l'Annexe A.

Cette classe indique que le principal s'est authentifié au moyen d'un certificat client, sécurisé avec le transport TLS.

```
<?xml version="1.0" encoding="UTF-8"?>

<xs:schema
targetNamespace="urn:oasis:names:tc:SAML:2.0:ac:classes:TLSClient"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns="urn:oasis:names:tc:SAML:2.0:ac:classes:TLSClient"
  finalDefault="extension"
  blockDefault="substitution"
  version="2.0">

  <xs:redefine schemaLocation="saml-schema-authn-context-types-2.0.xsd">

    <xs:annotation>
      <xs:documentation>
        Class identifier: urn:oasis:names:tc:SAML:2.0:ac:classes:TLSClient
        Document identifier: saml-schema-authn-context-sslcert-2.0
        Location: http://docs.oasis-open.org/security/saml/v2.0/
        Revision history:
          V2.0 (March, 2005):
            New authentication context class schema for SAML V2.0.
      </xs:documentation>
    </xs:annotation>

    <xs:complexType name="AuthnContextDeclarationBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthnContextDeclarationBaseType">
          <xs:sequence>
            <xs:element ref="Identification" minOccurs="0"/>
            <xs:element ref="TechnicalProtection" minOccurs="0"/>
            <xs:element ref="OperationalProtection" minOccurs="0"/>
            <xs:element ref="AuthnMethod"/>
            <xs:element ref="GoverningAgreements" minOccurs="0"/>
            <xs:element ref="Extension" minOccurs="0"
maxOccurs="unbounded"/>
          </xs:sequence>
          <xs:attribute name="ID" type="xs:ID" use="optional"/>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="AuthnMethodBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthnMethodBaseType">
          <xs:sequence>
            <xs:element ref="PrincipalAuthenticationMechanism"/>
            <xs:element ref="Authenticator"/>
            <xs:element ref="AuthenticatorTransportProtocol"
minOccurs="0"/>
            <xs:element ref="Extension" minOccurs="0"
maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="PrincipalAuthenticationMechanismType">
      <xs:complexContent>
        <xs:restriction base="PrincipalAuthenticationMechanismType">
          <xs:sequence>
            <xs:element ref="RestrictedPassword"/>
          </xs:sequence>
          <xs:attribute name="preauth" type="xs:integer" use="optional"/>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="AuthenticatorBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthenticatorBaseType">
```

```
          <xs:sequence>
            <xs:element ref="DigSig"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="PublicKeyType">
      <xs:complexContent>
        <xs:restriction base="PublicKeyType">
          <xs:attribute name="keyValidation" type="xs:anyURI"
fixed="urn:oasis:names:tc:SAML:2.0:ac:classes:X509"/>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="AuthenticatorTransportProtocolType">
      <xs:complexContent>
        <xs:restriction base="AuthenticatorTransportProtocolType">
          <xs:sequence>
            <xs:choice>
              <xs:element ref="SSL"/>
              <xs:element ref="WTLS"/>
            </xs:choice>
            <xs:element ref="Extension" minOccurs="0"
maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

  </xs:redefine>

</xs:schema>
```

NOTE – L'utilisation de SSL est présentée à l'Appendice IV.

12.3.4.24 TimeSyncToken

URI: `urn:oasis:names:tc:SAML:2.0:ac:classes:TimeSyncToken`

Cet URI est aussi utilisé comme espace de nom cible dans le schéma de classe de contexte d'authentification de l'Annexe A.

La classe TimeSyncToken est applicable lorsqu'un principal s'authentifie par un jeton de synchronisation temporelle.

```
<?xml version="1.0" encoding="UTF-8"?>

<xs:schema
targetNamespace="urn:oasis:names:tc:SAML:2.0:ac:classes:TimeSyncToken"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns="urn:oasis:names:tc:SAML:2.0:ac:classes:TimeSyncToken"
  finalDefault="extension"
  blockDefault="substitution"
  version="2.0">

  <xs:redefine schemaLocation="saml-schema-authn-context-types-2.0.xsd">

    <xs:annotation>
      <xs:documentation>
        Class identifier:
urn:oasis:names:tc:SAML:2.0:ac:classes:TimeSyncToken
        Document identifier: saml-schema-authn-context-timesync-2.0
        Location: http://docs.oasis-open.org/security/saml/v2.0/
        Revision history:
          V2.0 (March, 2005):
            New authentication context class schema for SAML V2.0.
      </xs:documentation>
    </xs:annotation>

    <xs:complexType name="AuthnContextDeclarationBaseType">
```

```
      <xs:complexContent>
        <xs:restriction base="AuthnContextDeclarationBaseType">
          <xs:sequence>
            <xs:element ref="Identification" minOccurs="0"/>
            <xs:element ref="TechnicalProtection" minOccurs="0"/>
            <xs:element ref="OperationalProtection" minOccurs="0"/>
            <xs:element ref="AuthnMethod"/>
            <xs:element ref="GoverningAgreements" minOccurs="0"/>
            <xs:element ref="Extension" minOccurs="0"
maxOccurs="unbounded"/>
          </xs:sequence>
          <xs:attribute name="ID" type="xs:ID" use="optional"/>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="AuthnMethodBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthnMethodBaseType">
          <xs:sequence>
            <xs:element ref="PrincipalAuthenticationMechanism"
minOccurs="0"/>
            <xs:element ref="Authenticator"/>
            <xs:element ref="AuthenticatorTransportProtocol"
minOccurs="0"/>
            <xs:element ref="Extension" minOccurs="0"
maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="PrincipalAuthenticationMechanismType">
      <xs:complexContent>
        <xs:restriction base="PrincipalAuthenticationMechanismType">
          <xs:sequence>
            <xs:element ref="Token"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="TokenType">
      <xs:complexContent>
        <xs:restriction base="TokenType">
          <xs:sequence>
            <xs:element ref="TimeSyncToken"/>
            <xs:element ref="Extension" minOccurs="0"
maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="TimeSyncTokenType">
      <xs:complexContent>
        <xs:restriction base="TimeSyncTokenType">
          <xs:attribute name="DeviceType" use="required">
            <xs:simpleType>
              <xs:restriction base="DeviceTypeType">
                <xs:enumeration value="hardware"/>
              </xs:restriction>
            </xs:simpleType>
          </xs:attribute>

          <xs:attribute name="SeedLength" use="required">
            <xs:simpleType>
              <xs:restriction base="xs:integer">
                <xs:minInclusive value="64"/>
              </xs:restriction>
            </xs:simpleType>
```

```
          </xs:attribute>

          <xs:attribute name="DeviceInHand" use="required">
            <xs:simpleType>
              <xs:restriction base="booleanType">
                <xs:enumeration value="true"/>
              </xs:restriction>
            </xs:simpleType>
          </xs:attribute>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

  </xs:redefine>

</xs:schema>
```

12.3.4.25 Unspecified

URI: `urn:oasis:names:tc:SAML:2.0:ac:classes:Unspecified`

La classe Unspecified indique que l'authentification a été effectuée par des moyens non spécifiés.

13 Exigences de conformité pour SAML

Le présent paragraphe décrit les caractéristiques qui sont obligatoires et celles qui sont facultatives pour les implémentations qui revendiquent la conformité à SAML.

La présente Recommandation définit un certain nombre de profils désignés. Chaque profil (autres que les profils d'attribut) décrit les détails des flux de message SAML et peut aussi être vu comme une fonctionnalité indivisible qui pourrait être implémentée par un composant logiciel. L'implémentation d'un profil implique l'utilisation d'une liaison pour chaque échange de messages inclus dans le profil. Une liaison peut être vue comme une technique d'implémentation spécifique pour réaliser un échange de messages.

Le présent paragraphe énumère tous les différents profils définis dans la présente Recommandation. Pour chaque profil, le flux de messages SAML V2.0 pertinent est cité, et pour chaque flux de messages, l'ensemble des liaisons possibles est aussi décrit. La combinaison du profil, de l'échange de messages et d'une liaison choisie est appelée une *caractéristique* SAML V2.0.

Le présent paragraphe décrit aussi la matrice de conformité pour SAML V2.0. Un certain nombre de *modes opérationnels* ou rôles différents sont identifiés. La matrice de conformité décrit l'ensemble des caractéristiques qui doit être implémenté par chaque mode opérationnel.

13.1 Profils SAML et implémentations possibles

Le Tableau 1 énumère tous les profils définis pour SAML. Pour chaque profil, on décrit aussi les flux de message de protocole qui s'y trouvent. Pour chaque flux de message, une liste des liaisons pertinentes est donnée dans la dernière colonne.

Tableau 1/X.1141 – Implémentations possibles

Profil	Flux de messages	Liaison
SSO de la toile	`<AuthnRequest>` de SP à IdP	HTTP Redirect
		HTTP POST
		HTTP Artifact
	IdP `<Response>` à SP	HTTP POST
		HTTP Artifact
SSO client/mandataire amélioré	ECP à SP, SP à ECP à IdP	PAOS
	IdP à ECP à SP, SP à ECP	PAOS
Découverte d'identité du fournisseur	Installe un mouchard	HTTP
	Installe un mouchard	HTTP

Tableau 1/X.1141 – Implémentations possibles

Profil	Flux de messages	Liaison
Terminaison de session unique	<LogoutRequest>	HTTP Redirect
		HTTP POST
		HTTP Artifact
		SOAP
	<LogoutResponse>	HTTP Redirect
		HTTP POST
		HTTP Artifact
		SOAP
Gestion d'identifiant de nom	<ManageNameIDRequest>	HTTP Redirect
		HTTP POST
		HTTP Artifact
		SOAP
	<ManageNameIDResponse>	HTTP Redirect
		SOAP
Résolution d'artifice	<ArtifactResolve>, <ArtifactResponse>	SOAP
Interrogation d'authentification	<AuthnQuery>, <Response>	SOAP
Interrogation d'attribut	<AttributeQuery>, <Response>	SOAP
Interrogation de décision d'autorisation	<AuthzDecisionQuery>, <Response>	SOAP
Demande d'assertion par identifiant	<AssertionIDRequest>, <Response>	SOAP
Transposition d'identifiant de nom	<NameIDMappingRequest>, <NameIDMappingResponse>	SOAP
Liaison d'URI SAML	GET, HTTP Response	HTTP
Profil d'attribut d'UUID		
Profil d'attribut PAC DCE		
Profil d'attribut X.500		
Profil d'attribut XACML		
Métadonnées	Consommation	
	Échange	

13.2 Conformité

Ce paragraphe décrit les exigences de conformité technique pour SAML V2.0.

13.2.1 Modes opérationnels

La présente Recommandation utilise la phrase "mode opérationnel" pour décrire un rôle que peut jouer un composant logiciel en se conformant à SAML. Les modes opérationnels sont les suivants:

- IdP – Fournisseur d'identité
- IdP Lite – Fournisseur d'identité léger
- SP – Fournisseur de service
- SP Lite – Fournisseur de service léger
- ECP – client/mandataire amélioré
- Autorité d'attribut SAML
- Autorité de décision d'autorisation SAML
- Autorité d'authentification SAML
- Demandeur SAML

13.2.2 Matrice des caractéristiques

Les matrices suivantes (voir Tableau 2) identifient des ensembles uniques d'exigences de conformité au moyen d'un triplet tiré du Tableau 1 de la forme: profil, message(s), liaison. Le composant de message n'est pas toujours inclus lorsqu'il est évident d'après le contexte.

Tableau 2/X.1141 – Matrice des caractéristiques

Caractéristique	IdP	IdP lite	SP	SP lite	ECP
Web SSO, `<AuthnRequest>`, HTTP redirect	Doit	Doit	Doit	Doit	N/A
Web SSO, `<Response>`, HTTP POST	Doit	Doit	Doit	Doit	N/A
Web SSO, `<Response>`, HTTP artifact	Doit	Doit	Doit	Doit	N/A
Résolution d'artifice, SOAP	Doit	Doit	Doit	Doit	N/A
SSO client/mandataire amélioré, PAOS	Doit	Doit	Doit	Doit	Doit
Gestion d'identifiant de nom, HTTP redirect (initié par IdP)	Doit	Ne doit pas	Doit	Ne doit pas	N/A
Gestion d'identifiant de nom, SOAP (initié par IdP)	Doit	Ne doit pas	Facultatif	Ne doit pas	N/A
Gestion d'identifiant de nom, HTTP redirect NOTE (informative) : PE11 (voir OASIS PE:2006) suggère d'ajouter (initié par SP)	Doit	Ne doit pas	Doit	Ne doit pas	N/A
Gestion d'identifiant de nom, SOAP (initié par SP)	Doit	Ne doit pas	Facultatif	Ne doit pas	N/A
Terminaison de session unique (initié par IdP) – HTTP redirect	Doit	Doit	Doit	Doit	N/A
Terminaison de session unique (initié par IdP) – SOAP	Doit	Facultatif	Doit	Facultatif	N/A
Terminaison de session unique (initié par SP) – HTTP redirect	Doit	Doit	Doit	Doit	N/A
Terminaison de session unique (initié par SP) – SOAP	Doit	Facultatif	Doit	Facultatif	N/A
Découverte de fournisseur d'identité (mouchard)	Doit	Doit	Facultatif	Facultatif	N/A

NOTE 1 (informative) – PE16 (voir OASIS PE:2006) suggère de remplacer N/A par "Facultatif " à la dernière rangée de la dernière colonne du Tableau 2.

NOTE 2 (informative) – PE25 (voir OASIS PE:2006) suggère d'ajouter ce qui suit à la fin du Tableau 2:

Caractéristique	IdP	IdP Lite	SP	SP Lite	ECP
Structures de métadonnées	Facultatif	Facultatif	Facultatif	Facultatif	N/A
Interoperation de métadonnées	Facultatif	Facultatif	Facultatif	Facultatif	N/A

NOTE 3 (informative) – PE29 (voir OASIS PE:2006) suggère d'ajouter ce qui suit à la fin du Tableau 2:

Caractéristique	IdP	IdP Lite	SP	SP Lite	ECP
Demande d'identifiant d'assertion	Facultatif	N/A	N/A	N/A	N/A
Liaison d'URI SAML	Facultatif	N/A	N/A	N/A	N/A

Le Tableau 3 récapitule les modes opérationnels qui étendent les modes IdP ou SP définis ci-dessus. Il faut les comprendre comme une combinaison d'un mode IdP ou SP à partir du tableau ci-dessus avec l'ensemble de caractéristiques étendues correspondant ci-dessous.

Tableau 3/X.1141 – IdP, SP étendus

Caractéristique	IdP étendu	SP étendu
Mandataire fournisseur d'identité	Doit	Doit
Transposition d'identifiant de nom, SOAP	Doit	Doit

Le Tableau 4 récapitule les exigences de conformité pour les autorités et demandeurs SAML.

Tableau 4/X.1141 – Matrice des autorité et demandeurs SAML

Caractéristique	Autorité d'authentification SAML	Autorité d'attribut SAML	Autorité de décision d'autorisation SAML	demandeur SAML
Interrogation d'authentication, SOAP	Doit	Facultatif	Facultatif	Facultatif
Interrogation d'attribut, SOAP	Facultatif	Doit	Facultatif	Facultatif
Interrogation de décision d'autorisation, SOAP	Facultatif	Facultatif	Doit	Facultatif
Demande d'assertion par identifiant, SOAP	Doit	Doit	Doit	Facultatif
Liaison d'URI SAML	Doit	Doit	Doit	Facultatif

NOTE 4 (informative) – PE25 et PE42 (voir OASIS PE:2006) suggèrent de modifier le Tableau 4 ci-dessus comme suit:

Caractéristique	Autorité d'authentification SAML	Autorité d'attribut SAML	Autorité de décision d'autorisation SAML	demandeur SAML
Interrogation d'authentification, SOAP	Doit	N/A	N/A	Facultatif
Interrogation d'attribut, SOAP	N/A	Doit	N/A	Facultatif
Interrogation de décision d'autorisation, SOAP	N/A	N/A	Doit	Facultatif
Demande d'assertion par identifiant, SOAP	Doit	Doit	Doit	Facultatif
Liaison d'URI SAML	Doit	Doit	Doit	Facultatif
Structures de métadonnées	Facultatif	Facultatif	Facultatif	Facultatif
Interopération de métadonnées	Facultatif	Facultatif	Facultatif	Facultatif

13.2.3 Implémentation des identifiants définis par SAML

Tous les modes opérationnels pertinents doivent implémenter les identifiants définis par SAML suivants:

- tous les identifiants de format de nom d'attribut définis au § 8;
- tous les identifiants de format d'identifiant de nom définis au § 8.

Les implémentations conformes à SAML doivent permettre l'utilisation de toutes les constantes d'identifiant (voir aux § 8.1 et 8.2) lors de la production et de la consommation des messages SAML. Les producteurs de message SAML doivent être capables de créer des messages et les consommateurs de message SAML doivent être capables de traiter les messages avec toutes les constantes définies dans ces paragraphes.

Les identifiants de nom persistants et les identifiants de nom transitoires définissent des règles de traitement normatives pour le producteur de tels identifiants. Toutes les règles de traitement normatives doivent être prises en charge par les implémentations conformes. Les identifiants restants ne spécifient pas de règles de traitement normatives. Et donc, la génération et la consommation de ces identifiants n'est significative que lorsque les parties génératrice et consommatrice ont un accord défini de façon externe sur l'interprétation sémantique des identifiants.

NOTE – Dans ce contexte, "traiter" signifie que l'implémentation doit réussir à analyser et à traiter l'identifiant sans échouer ni retourner d'erreur. La façon dont l'implémentation traite les identifiants une fois qu'ils ont été traités à ce niveau est en dehors du domaine d'application de la présente Recommandation.

Une implémentation SAML peut fournir les facilités décrites ci-dessus par une prise en charge directe des identifiants par l'implémentation ou par l'utilisation d'interfaces de programmation adaptées. Les interfaces fournies à cette fin doivent permettre à l'implémentation SAML d'avoir les extensions de programme nécessaires pour traiter tous les identifiants qui ne sont pas traités d'origine par l'implémentation.

13.2.4 Implémentation des éléments chiffrés

Tous les modes opérationnels pertinents doivent être capables de traiter ou générer les éléments chiffrés suivants dans tout contexte où ils sont nécessaires pour traiter ou générer les éléments non chiffrés correspondants, à savoir `<saml:NameID>`, `<saml:Assertion>`, ou `<saml:Attribute>`:

- `<saml:EncryptedID>`
- `<saml:EncryptedAssertion>`
- `<saml:EncryptedAttribute>`

13.2.5 Modèles de sécurité pour les liaisons SOAP et URI

L'implémentation des modèles de sécurité suivants est obligatoire pour tous les profils qui utilisent la liaison SOAP ainsi que pour la liaison d'URI SAML. Les autorités et demandeurs SAML doivent implémenter les méthodes d'authentification suivantes:

- pas d'authentification client ou serveur;
- authentification HTTP de base avec et sans TLS 1.0. Le demandeur SAML doit envoyer préalablement l'en-tête d'autorisation avec la demande initiale;
- authentification de serveur HTTP sur TLS 1.0 avec certificat côté serveur;
- authentification mutuelle HTTP sur TLS 1.0 avec certificat côté serveur et côté client.

Si une autorité SAML utilise TLS 1.0, elle doit utiliser un certificat côté serveur.

NOTE 1 (informative) – PE25 (voir OASIS PE:2006) suggère d'ajouter un nouveau paragraphe sur les structures de métadonnées comme suit:

Les implémentations qui revendiquent la conformité à SAML peuvent déclarer la conformité aux métadonnées SAML de chaque mode opérationnel en choisissant l'option Structures de métadonnées. Par rapport à chaque mode opérationnel, une telle conformité entraine ce qui suit:

L'implémentation des métadonnées SAML conformément au format extensible de métadonnées SAML dans tous les cas où un homologue interopérant a l'option, comme établi dans les spécifications SAML, de dépendre de l'existence de métadonnées SAML. Le choix de l'option Structures de métadonnées a pour effet d'exiger que de telles métadonnés soient disponibles pour l'homologue interopérant. La caractéristique de métadonnées interopérantes, décrite ci-dessous, fournit un moyen de satisfaire à cette exigence.

La référence, la consommation, et l'adhésion aux métadonnées SAML, conformément à la présente Recommandation, d'un homologue interopérant lorsque les métadonnées connues pertinentes pour cet homologue, pour l'opération en cause, et pour l'échange en cours, sont arrivées à expiration ou ne sont plus valides dans la mémoire cache, sont possibles, pourvu que les métadonnées soient disponibles et ne soient pas interdites par la politique ou par l'opération en cause et cet échange spécifique.

NOTE 2 (informative) – PE25 (voir OASIS PE:2006) suggère d'ajouter un nouveau paragraphe sur les métadonnées interopérantes comme suit:

Le choix de l'option métadonnées interopérantes exige que l'implémentation offre, en plus de tout autre mécanisme, le mécanisme bien connu de localisation, publication et résolution décrit au § 9 de SAML, sur les métadonnées.

13.3 Signature numérique XML et chiffrement XML

SAML V2.0 utilise XML Signature pour implémenter la fonction de signature et de chiffrement XML pour la protection de l'intégrité et l'authentification de la source. SAML V2.0 utilise XML Encryption pour implémenter la confidentialité, y compris les identifiants chiffrés, les assertions chiffrées, et les attributs chiffrés.

13.3.1 Algorithmes de signature XML

XML Signature, 6.1, du W3C rend obligatoire l'utilisation de ce qui suit:

- Digest: SHA-1;
- MAC: HMAC-SHA1;
- Canonisation XML: CanonicalXML (sans commentaires);
- Transform: signature enveloppée.

Ils doivent donc être appliqués par les implémentations SAML V2.0 conformes.

De plus, pour permettre l'interopérabilité, ce qui suit doit être appliqué par les implémentations conformes à SAML V2.0:

- Signature: RSAwithSHA1 (recommandé par W3C Signature, nécessaire pour l'interopérabilité).

Bien que XML Signature rende obligatoire l'algorithme de signature DSAwithSHA1, il n'est pas exigé par SAML V2.0, mais recommandé.

NOTE – L'Institut national [USA] des normes et des techniques encourage maintenant l'utilisation de SHA-256 (Algorithme de hachage sécurisé à clés codées de 256 bits) à la place de SHA-1.

13.3.2 Algorithmes de chiffrement XML

- Les paragraphes 5.2.1 et 5.2.2 de XML Encryption du W3C, rendent obligatoire l'utilisation des algorithmes suivants: chiffrement de bloc: Triple DES, AES-128, AES-256.
- Transport des clés: RSA-v1.5, RSA-OAEP.

Les algorithmes ci-dessus doivent donc être appliqués par les implémentations conformes à SAML V2.0.

13.4 Utilisation de TLS 1.0

Dans toute utilisation de TLS 1.0 par SAML V2.0, les serveurs doivent s'authentifier auprès des clients en utilisant un certificat X.509 v3. Le client doit établir l'identité du serveur sur la base du contenu du certificat (normalement par l'examen du champ DN du sujet du certificat).

13.4.1 Liaison SOAP et URI de SAML

Les implémentations à capacité TLS doivent implémenter la suite de chiffrement TLS_RSA_WITH_3DES_EDE_CBC_SHA et peuvent implémenter la suite de chiffrement TLS_RSA_AES_128_CBC_SHA.

Les implémentations à capacité FIPS TLS doivent implémenter la suite de chiffrement TLS_RSA_FIPS_WITH_3DES_EDE_CBC_SHA correspondante et peuvent implémenter la suite de chiffrement TLS_RSA_FIPS_AES_128_CBC_SHA correspondante.

13.4.2 Profils SSO de la toile de SAML

Les implémentations à capacité TLS doivent implémenter la suite de chiffrement TLS_RSA_WITH_3DES_EDE_CBC_SHA (voir la RFC 2246 de l'IETF).

Annexe A

Schémas SAML

La présente annexe fait partie intégrante de la présente Recommandation. Elle donne la liste des schémas SAML exigés.

A.1 Schéma SAML Assertion

Listing du schéma SAML Assertion:

```
<?xml version="1.0" encoding="US-ASCII"?>
<schema
    targetNamespace="urn:oasis:names:tc:SAML:2.0:assertion"
    xmlns="http://www.w3.org/2001/XMLSchema"
    xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion"
    xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
    xmlns:xenc="http://www.w3.org/2001/04/xmlenc#"
    elementFormDefault="unqualified"
    attributeFormDefault="unqualified"
    blockDefault="substitution"
    version="2.0">
    <import namespace="http://www.w3.org/2000/09/xmldsig#"
        schemaLocation="http://www.w3.org/TR/2002/REC-xmldsig-core-
20020212/xmldsig-core-schema.xsd"/>
    <import namespace="http://www.w3.org/2001/04/xmlenc#"
```

```
        schemaLocation="http://www.w3.org/TR/2002/REC-xmlenc-core-20021210/xenc-
schema.xsd"/>
    <annotation>
        <documentation>
            Document identifier: saml-schema-assertion-2.0
            Location: http://docs.oasis-open.org/security/saml/v2.0/
            Revision history:
            V1.0 (November, 2002):
              Initial Standard Schema.
            V1.1 (September, 2003):
              Updates within the same V1.0 namespace.
            V2.0 (March, 2005):
              New assertion schema for SAML V2.0 namespace.
        </documentation>
    </annotation>
    <attributeGroup name="IDNameQualifiers">
        <attribute name="NameQualifier" type="string" use="optional"/>
        <attribute name="SPNameQualifier" type="string" use="optional"/>
    </attributeGroup>
    <element name="BaseID" type="saml:BaseIDAbstractType"/>
    <complexType name="BaseIDAbstractType" abstract="true">
        <attributeGroup ref="saml:IDNameQualifiers"/>
    </complexType>
    <element name="NameID" type="saml:NameIDType"/>
    <complexType name="NameIDType">
        <simpleContent>
            <extension base="string">
                <attributeGroup ref="saml:IDNameQualifiers"/>
                <attribute name="Format" type="anyURI" use="optional"/>
                <attribute name="SPProvidedID" type="string" use="optional"/>
            </extension>
        </simpleContent>
    </complexType>
    <complexType name="EncryptedElementType">
        <sequence>
            <element ref="xenc:EncryptedData"/>
            <element ref="xenc:EncryptedKey" minOccurs="0"
maxOccurs="unbounded"/>
        </sequence>
    </complexType>
    <element name="EncryptedID" type="saml:EncryptedElementType"/>
    <element name="Issuer" type="saml:NameIDType"/>
    <element name="AssertionIDRef" type="NCName"/>
    <element name="AssertionURIRef" type="anyURI"/>
    <element name="Assertion" type="saml:AssertionType"/>
    <complexType name="AssertionType">
        <sequence>
            <element ref="saml:Issuer"/>
            <element ref="ds:Signature" minOccurs="0"/>
            <element ref="saml:Subject" minOccurs="0"/>
            <element ref="saml:Conditions" minOccurs="0"/>
            <element ref="saml:Advice" minOccurs="0"/>
            <choice minOccurs="0" maxOccurs="unbounded">
                <element ref="saml:Statement"/>
                <element ref="saml:AuthnStatement"/>
                <element ref="saml:AuthzDecisionStatement"/>
                <element ref="saml:AttributeStatement"/>
            </choice>
        </sequence>
        <attribute name="Version" type="string" use="required"/>
        <attribute name="ID" type="ID" use="required"/>
        <attribute name="IssueInstant" type="dateTime" use="required"/>
    </complexType>
    <element name="Subject" type="saml:SubjectType"/>
    <complexType name="SubjectType">
        <choice>
            <sequence>
                <choice>
                    <element ref="saml:BaseID"/>
                    <element ref="saml:NameID"/>
                    <element ref="saml:EncryptedID"/>
```

```
                </choice>
                <element ref="saml:SubjectConfirmation" minOccurs="0"
maxOccurs="unbounded"/>
            </sequence>
            <element ref="saml:SubjectConfirmation" maxOccurs="unbounded"/>
        </choice>
    </complexType>
    <element name="SubjectConfirmation" type="saml:SubjectConfirmationType"/>
    <complexType name="SubjectConfirmationType">
        <sequence>
            <choice minOccurs="0">
                <element ref="saml:BaseID"/>
                <element ref="saml:NameID"/>
                <element ref="saml:EncryptedID"/>
            </choice>
            <element ref="saml:SubjectConfirmationData" minOccurs="0"/>
        </sequence>
        <attribute name="Method" type="anyURI" use="required"/>
    </complexType>
    <element name="SubjectConfirmationData"
type="saml:SubjectConfirmationDataType"/>
    <complexType name="SubjectConfirmationDataType" mixed="true">
        <complexContent>
            <restriction base="anyType">
                <sequence>
                    <any namespace="##any" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
                </sequence>
                <attribute name="NotBefore" type="dateTime" use="optional"/>
                <attribute name="NotOnOrAfter" type="dateTime" use="optional"/>
                <attribute name="Recipient" type="anyURI" use="optional"/>
                <attribute name="InResponseTo" type="NCName" use="optional"/>
                <attribute name="Address" type="string" use="optional"/>
                <anyAttribute namespace="##other" processContents="lax"/>
            </restriction>
        </complexContent>
    </complexType>
    <complexType name="KeyInfoConfirmationDataType" mixed="false">
        <complexContent>
            <restriction base="saml:SubjectConfirmationDataType">
                <sequence>
                    <element ref="ds:KeyInfo" maxOccurs="unbounded"/>
                </sequence>
            </restriction>
        </complexContent>
    </complexType>
    <element name="Conditions" type="saml:ConditionsType"/>
    <complexType name="ConditionsType">
        <choice minOccurs="0" maxOccurs="unbounded">
            <element ref="saml:Condition"/>
            <element ref="saml:AudienceRestriction"/>
            <element ref="saml:OneTimeUse"/>
            <element ref="saml:ProxyRestriction"/>
        </choice>
        <attribute name="NotBefore" type="dateTime" use="optional"/>
        <attribute name="NotOnOrAfter" type="dateTime" use="optional"/>
    </complexType>
    <element name="Condition" type="saml:ConditionAbstractType"/>
    <complexType name="ConditionAbstractType" abstract="true"/>
    <element name="AudienceRestriction" type="saml:AudienceRestrictionType"/>
    <complexType name="AudienceRestrictionType">
        <complexContent>
            <extension base="saml:ConditionAbstractType">
                <sequence>
                    <element ref="saml:Audience" maxOccurs="unbounded"/>
                </sequence>
            </extension>
        </complexContent>
    </complexType>
    <element name="Audience" type="anyURI"/>
    <element name="OneTimeUse" type="saml:OneTimeUseType" />
```

```
    <complexType name="OneTimeUseType">
        <complexContent>
            <extension base="saml:ConditionAbstractType"/>
        </complexContent>
    </complexType>
    <element name="ProxyRestriction" type="saml:ProxyRestrictionType"/>
    <complexType name="ProxyRestrictionType">
    <complexContent>
        <extension base="saml:ConditionAbstractType">
            <sequence>
                <element ref="saml:Audience" minOccurs="0"
maxOccurs="unbounded"/>
            </sequence>
            <attribute name="Count" type="nonNegativeInteger" use="optional"/>
        </extension>
   </complexContent>
    </complexType>
    <element name="Advice" type="saml:AdviceType"/>
    <complexType name="AdviceType">
        <choice minOccurs="0" maxOccurs="unbounded">
            <element ref="saml:AssertionIDRef"/>
            <element ref="saml:AssertionURIRef"/>
            <element ref="saml:Assertion"/>
            <element ref="saml:EncryptedAssertion"/>
            <any namespace="##other" processContents="lax"/>
        </choice>
    </complexType>
    <element name="EncryptedAssertion" type="saml:EncryptedElementType"/>
    <element name="Statement" type="saml:StatementAbstractType"/>
    <complexType name="StatementAbstractType" abstract="true"/>
    <element name="AuthnStatement" type="saml:AuthnStatementType"/>
    <complexType name="AuthnStatementType">
        <complexContent>
            <extension base="saml:StatementAbstractType">
                <sequence>
                    <element ref="saml:SubjectLocality" minOccurs="0"/>
                    <element ref="saml:AuthnContext"/>
                </sequence>
                <attribute name="AuthnInstant" type="dateTime" use="required"/>
                <attribute name="SessionIndex" type="string" use="optional"/>
                <attribute name="SessionNotOnOrAfter" type="dateTime"
use="optional"/>
            </extension>
        </complexContent>
    </complexType>
    <element name="SubjectLocality" type="saml:SubjectLocalityType"/>
    <complexType name="SubjectLocalityType">
        <attribute name="Address" type="string" use="optional"/>
        <attribute name="DNSName" type="string" use="optional"/>
    </complexType>
    <element name="AuthnContext" type="saml:AuthnContextType"/>
    <complexType name="AuthnContextType">
        <sequence>
            <choice>
                <sequence>
                    <element ref="saml:AuthnContextClassRef"/>
                    <choice minOccurs="0">
                        <element ref="saml:AuthnContextDecl"/>
                        <element ref="saml:AuthnContextDeclRef"/>
                    </choice>
                </sequence>
                <choice>
                    <element ref="saml:AuthnContextDecl"/>
                    <element ref="saml:AuthnContextDeclRef"/>
                </choice>
            </choice>
            <element ref="saml:AuthenticatingAuthority" minOccurs="0"
maxOccurs="unbounded"/>
        </sequence>
    </complexType>
    <element name="AuthnContextClassRef" type="anyURI"/>
```

```
    <element name="AuthnContextDeclRef" type="anyURI"/>
    <element name="AuthnContextDecl" type="anyType"/>
    <element name="AuthenticatingAuthority" type="anyURI"/>
    <element name="AuthzDecisionStatement"
type="saml:AuthzDecisionStatementType"/>
    <complexType name="AuthzDecisionStatementType">
        <complexContent>
            <extension base="saml:StatementAbstractType">
                <sequence>
                    <element ref="saml:Action" maxOccurs="unbounded"/>
                    <element ref="saml:Evidence" minOccurs="0"/>
                </sequence>
                <attribute name="Resource" type="anyURI" use="required"/>
                <attribute name="Decision" type="saml:DecisionType"
use="required"/>
            </extension>
        </complexContent>
    </complexType>
    <simpleType name="DecisionType">
        <restriction base="string">
            <enumeration value="Permit"/>
            <enumeration value="Deny"/>
            <enumeration value="Indeterminate"/>
        </restriction>
    </simpleType>
    <element name="Action" type="saml:ActionType"/>
    <complexType name="ActionType">
        <simpleContent>
            <extension base="string">
                <attribute name="Namespace" type="anyURI" use="required"/>
            </extension>
        </simpleContent>
    </complexType>
    <element name="Evidence" type="saml:EvidenceType"/>
    <complexType name="EvidenceType">
        <choice maxOccurs="unbounded">
            <element ref="saml:AssertionIDRef"/>
            <element ref="saml:AssertionURIRef"/>
            <element ref="saml:Assertion"/>
            <element ref="saml:EncryptedAssertion"/>
        </choice>
    </complexType>
    <element name="AttributeStatement" type="saml:AttributeStatementType"/>
    <complexType name="AttributeStatementType">
        <complexContent>
            <extension base="saml:StatementAbstractType">
                <choice maxOccurs="unbounded">
                    <element ref="saml:Attribute"/>
                    <element ref="saml:EncryptedAttribute"/>
                </choice>
            </extension>
        </complexContent>
    </complexType>
    <element name="Attribute" type="saml:AttributeType"/>
    <complexType name="AttributeType">
        <sequence>
            <element ref="saml:AttributeValue" minOccurs="0"
maxOccurs="unbounded"/>
        </sequence>
        <attribute name="Name" type="string" use="required"/>
        <attribute name="NameFormat" type="anyURI" use="optional"/>
        <attribute name="FriendlyName" type="string" use="optional"/>
        <anyAttribute namespace="##other" processContents="lax"/>
    </complexType>
    <element name="AttributeValue" type="anyType" nillable="true"/>
    <element name="EncryptedAttribute" type="saml:EncryptedElementType"/>
</schema>
```

A.2 Schéma SAML Contexte d'authentification

Cette liste fournit le schéma SAML Contexte d'authentification:

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema
  targetNamespace="urn:oasis:names:tc:SAML:2.0:ac"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns="urn:oasis:names:tc:SAML:2.0:ac"
  blockDefault="substitution"
  version="2.0">

  <xs:annotation>
    <xs:documentation>
      Document identifier: saml-schema-authn-context-2.0
      Location: http://docs.oasis-open.org/security/saml/v2.0/
      Revision history:
        V2.0 (March, 2005):
          New core authentication context schema for SAML V2.0.
          This is just an include of all types from the Shema
          referred to in the include statement below.
    </xs:documentation>
  </xs:annotation>

  <xs:include schemaLocation="saml-schema-authn-context-types-2.0.xsd"/>

</xs:schema>
```

A.3 Schéma SAML de contexte d'authentification AuthenticatedTelephony

Schéma SAML de contexte d'authentification se rapportant à la téléphonie:

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema
targetNamespace="urn:oasis:names:tc:SAML:2.0:ac:classes:AuthenticatedTelephony"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns="urn:oasis:names:tc:SAML:2.0:ac:classes:AuthenticatedTelephony"
  finalDefault="extension"
  blockDefault="substitution"
  version="2.0">

  <xs:redefine schemaLocation="saml-schema-authn-context-types-2.0.xsd">

    <xs:annotation>
      <xs:documentation>
        Class identifier:
urn:oasis:names:tc:SAML:2.0:ac:classes:AuthenticatedTelephony
        Document identifier: saml-schema-authn-context-auth-telephony-2.0
        Location: http://docs.oasis-open.org/security/saml/v2.0/
        Revision history:
          V2.0 (March, 2005):
            New authentication_u99 context class schema for SAML V2.0.
      </xs:documentation>
    </xs:annotation>

    <xs:complexType name="AuthnContextDeclarationBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthnContextDeclarationBaseType">
          <xs:sequence>
            <xs:element ref="Identification" minOccurs="0"/>
            <xs:element ref="TechnicalProtection" minOccurs="0"/>
            <xs:element ref="OperationalProtection" minOccurs="0"/>
            <xs:element ref="AuthnMethod"/>
            <xs:element ref="GoverningAgreements" minOccurs="0"/>
            <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
          </xs:sequence>
          <xs:attribute name="ID" type="xs:ID" use="optional"/>
        </xs:restriction>
      </xs:complexContent>
```

```
    </xs:complexType>

    <xs:complexType name="AuthnMethodBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthnMethodBaseType">
          <xs:sequence>
            <xs:element ref="PrincipalAuthenticationMechanism" minOccurs="0"/>
            <xs:element ref="Authenticator"/>
            <xs:element ref="AuthenticatorTransportProtocol"/>
            <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="AuthenticatorBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthenticatorBaseType">
          <xs:sequence>
            <xs:element ref="Password"/>
            <xs:element ref="SubscriberLineNumber"/>
            <xs:element ref="UserSuffix"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="AuthenticatorTransportProtocolType">
      <xs:complexContent>
        <xs:restriction base="AuthenticatorTransportProtocolType">
          <xs:sequence>
            <xs:choice>
              <xs:element ref="PSTN"/>
              <xs:element ref="ISDN"/>
              <xs:element ref="ADSL"/>
            </xs:choice>
            <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

  </xs:redefine>

</xs:schema>
```

A.4 Schéma SAML du contexte d'authentification IP

Cette liste fournit le schéma SAML du contexte d'authentification spécifique de IP:

```
<?xml version="1.0" encoding="UTF-8"?>

<xs:schema
  targetNamespace="urn:oasis:names:tc:SAML:2.0:ac:classes:InternetProtocol"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns="urn:oasis:names:tc:SAML:2.0:ac:classes:InternetProtocol"
  finalDefault="extension"
  blockDefault="substitution"
  version="2.0">

  <xs:redefine schemaLocation="saml-schema-authn-context-types-2.0.xsd">

    <xs:annotation>
      <xs:documentation>
        Class identifier: urn:oasis:names:tc:SAML:2.0:ac:classes:InternetProtocol
        Document identifier: saml-schema-authn-context-ip-2.0
        Location: http://docs.oasis-open.org/security/saml/v2.0/
        Revision history:
          V2.0 (March, 2005):
```

```
            New authentication_u99 context class schema for SAML V2.0.
        </xs:documentation>
    </xs:annotation>

    <xs:complexType name="AuthnContextDeclarationBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthnContextDeclarationBaseType">
          <xs:sequence>
            <xs:element ref="Identification" minOccurs="0"/>
            <xs:element ref="TechnicalProtection" minOccurs="0"/>
            <xs:element ref="OperationalProtection" minOccurs="0"/>
            <xs:element ref="AuthnMethod"/>
            <xs:element ref="GoverningAgreements" minOccurs="0"/>
            <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
          </xs:sequence>
          <xs:attribute name="ID" type="xs:ID" use="optional"/>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="AuthnMethodBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthnMethodBaseType">
          <xs:sequence>
            <xs:element ref="PrincipalAuthenticationMechanism" minOccurs="0"/>
            <xs:element ref="Authenticator"/>
            <xs:element ref="AuthenticatorTransportProtocol" minOccurs="0"/>
            <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="AuthenticatorBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthenticatorBaseType">
          <xs:sequence>
            <xs:element ref="IPAddress"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

  </xs:redefine>

</xs:schema>
```

A.5 Schéma SAML du contexte d'authentification IPPWord

Cette liste fournit le schéma de contexte d'authentification SAML protocole Internet avec mot de passe (IPPWord).

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema
targetNamespace="urn:oasis:names:tc:SAML:2.0:ac:classes:InternetProtocolPassword"
  xmlns:ac="urn:oasis:names:tc:SAML:2.0:ac"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns="urn:oasis:names:tc:SAML:2.0:ac:classes:InternetProtocolPassword"
  finalDefault="extension"
  blockDefault="substitution"
  version="2.0">

  <xs:redefine schemaLocation="saml-schema-authn-context-types-2.0.xsd">

    <xs:annotation>
      <xs:documentation>
        Class identifier:
urn:oasis:names:tc:SAML:2.0:ac:classes:InternetProtocolPassword
        Document identifier: saml-schema-authn-context-ippword-2.0
        Location: http://docs.oasis-open.org/security/saml/v2.0/
```

```
        Revision history:
          V2.0 (March, 2005):
            New authentication_u99 context class schema for SAML V2.0.
      </xs:documentation>
    </xs:annotation>

    <xs:complexType name="AuthnContextDeclarationBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthnContextDeclarationBaseType">
          <xs:sequence>
            <xs:element ref="Identification" minOccurs="0"/>
            <xs:element ref="TechnicalProtection" minOccurs="0"/>
            <xs:element ref="OperationalProtection" minOccurs="0"/>
            <xs:element ref="AuthnMethod"/>
            <xs:element ref="GoverningAgreements" minOccurs="0"/>
            <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
          </xs:sequence>
          <xs:attribute name="ID" type="xs:ID" use="optional"/>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="AuthnMethodBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthnMethodBaseType">
          <xs:sequence>
            <xs:element ref="PrincipalAuthenticationMechanism" minOccurs="0"/>
            <xs:element ref="Authenticator"/>
            <xs:element ref="AuthenticatorTransportProtocol" minOccurs="0"/>
            <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="AuthenticatorBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthenticatorBaseType">
          <xs:sequence>
            <xs:element ref="Password"/>
            <xs:element ref="IPAddress"/>
            <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

  </xs:redefine>

</xs:schema>
```

A.6 Schéma SAML du contexte d'authentification Kerberos

Cette liste fournit le schéma d'authentification Kerberos de SAML.

```
<?xml version="1.0" encoding="UTF-8"?>

<xs:schema targetNamespace="urn:oasis:names:tc:SAML:2.0:ac:classes:Kerberos"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns="urn:oasis:names:tc:SAML:2.0:ac:classes:Kerberos"
  finalDefault="extension"
  blockDefault="substitution"
  version="2.0">

  <xs:redefine schemaLocation="saml-schema-authn-context-types-2.0.xsd">

    <xs:annotation>
      <xs:documentation>
        Class identifier: urn:oasis:names:tc:SAML:2.0:ac:classes:Kerberos
```

```
        Document identifier: saml-schema-authn-context-kerberos-2.0
        Location: http://docs.oasis-open.org/security/saml/v2.0/
        Revision history:
          V2.0 (March, 2005):
            New authentication_u99 context class schema for SAML V2.0.
      </xs:documentation>
    </xs:annotation>

    <xs:complexType name="AuthnContextDeclarationBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthnContextDeclarationBaseType">
          <xs:sequence>
            <xs:element ref="Identification" minOccurs="0"/>
            <xs:element ref="TechnicalProtection" minOccurs="0"/>
            <xs:element ref="OperationalProtection" minOccurs="0"/>
            <xs:element ref="AuthnMethod"/>
            <xs:element ref="GoverningAgreements" minOccurs="0"/>
            <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
          </xs:sequence>
          <xs:attribute name="ID" type="xs:ID" use="optional"/>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="AuthnMethodBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthnMethodBaseType">
          <xs:sequence>
            <xs:element ref="PrincipalAuthenticationMechanism"/>
            <xs:element ref="Authenticator"/>
            <xs:element ref="AuthenticatorTransportProtocol" minOccurs="0"/>
            <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="PrincipalAuthenticationMechanismType">
      <xs:complexContent>
        <xs:restriction base="PrincipalAuthenticationMechanismType">
          <xs:sequence>
            <xs:element ref="RestrictedPassword"/>
          </xs:sequence>
          <xs:attribute name="preauth" type="xs:integer" use="optional"/>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="AuthenticatorBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthenticatorBaseType">
          <xs:sequence>
            <xs:element ref="SharedSecretChallengeResponse"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="SharedSecretChallengeResponseType">
      <xs:complexContent>
        <xs:restriction base="SharedSecretChallengeResponseType">
          <xs:attribute name="method" type="xs:anyURI"
fixed="urn:oasis:names:tc:SAML:2.0:ac:classes:Kerberos"/>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

  </xs:redefine>
</xs:schema>
```

A.7 Schéma SAML du contexte d'authentification MobileOneFactor-reg

Cette liste contient le schéma de classe de contexte SAML pour MobileOneFactorContract enregistré.

```
<?xml version="1.0" encoding="UTF-8"?>

<xs:schema
targetNamespace="urn:oasis:names:tc:SAML:2.0:ac:classes:MobileOneFactorContract"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns="urn:oasis:names:tc:SAML:2.0:ac:classes:MobileOneFactorContract"
  finalDefault="extension"
  blockDefault="substitution"
  version="2.0">

  <xs:redefine schemaLocation="saml-schema-authn-context-types-2.0.xsd">

    <xs:annotation>
      <xs:documentation>
        Class identifier:
urn:oasis:names:tc:SAML:2.0:ac:classes:MobileOneFactorContract
        Document identifier: saml-schema-authn-context-mobileonefactor-reg-2.0
        Location: http://docs.oasis-open.org/security/saml/v2.0/
        Revision history:
          V2.0 (March, 2005):
            New authentication_u99 context class schema for SAML V2.0.
      </xs:documentation>
    </xs:annotation>

    <xs:complexType name="AuthnContextDeclarationBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthnContextDeclarationBaseType">
          <xs:sequence>
            <xs:element ref="Identification" minOccurs="0"/>
            <xs:element ref="TechnicalProtection" minOccurs="0"/>
            <xs:element ref="OperationalProtection" minOccurs="0"/>
            <xs:element ref="AuthnMethod"/>
            <xs:element ref="GoverningAgreements" minOccurs="0"/>
            <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
          </xs:sequence>
          <xs:attribute name="ID" type="xs:ID" use="optional"/>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="AuthnMethodBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthnMethodBaseType">
          <xs:sequence>
            <xs:element ref="PrincipalAuthenticationMechanism" minOccurs="0"/>
            <xs:element ref="Authenticator"/>
            <xs:element ref="AuthenticatorTransportProtocol" minOccurs="0"/>
            <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="AuthenticatorBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthenticatorBaseType">
          <xs:sequence>
            <xs:choice>
              <xs:element ref="DigSig"/>
              <xs:element ref="ZeroKnowledge"/>
              <xs:element ref="SharedSecretChallengeResponse"/>
              <xs:element ref="SharedSecretDynamicPlaintext"/>
              <xs:element ref="AsymmetricDecryption"/>
              <xs:element ref="AsymmetricKeyAgreement"/>
            </xs:choice>
            <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
```

```
      </xs:sequence>
    </xs:restriction>
  </xs:complexContent>
</xs:complexType>

<xs:complexType name="AuthenticatorTransportProtocolType">
  <xs:complexContent>
    <xs:restriction base="AuthenticatorTransportProtocolType">
      <xs:sequence>
        <xs:choice>
          <xs:element ref="SSL"/>
          <xs:element ref="MobileNetworkNoEncryption"/>
          <xs:element ref="MobileNetworkRadioEncryption"/>
          <xs:element ref="MobileNetworkEndToEndEncryption"/>
          <xs:element ref="WTLS"/>
        </xs:choice>
        <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
      </xs:sequence>
    </xs:restriction>
  </xs:complexContent>
</xs:complexType>

<xs:complexType name="OperationalProtectionType">
  <xs:complexContent>
    <xs:restriction base="OperationalProtectionType">
      <xs:sequence>
        <xs:element ref="SecurityAudit"/>
        <xs:element ref="DeactivationCallCenter"/>
        <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
      </xs:sequence>
    </xs:restriction>
  </xs:complexContent>
</xs:complexType>

<xs:complexType name="TechnicalProtectionBaseType">
  <xs:complexContent>
    <xs:restriction base="TechnicalProtectionBaseType">
      <xs:sequence>
        <xs:choice>
          <xs:element ref="PrivateKeyProtection"/>
          <xs:element ref="SecretKeyProtection"/>
        </xs:choice>
        <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
      </xs:sequence>
    </xs:restriction>
  </xs:complexContent>
</xs:complexType>

<xs:complexType name="PrivateKeyProtectionType">
  <xs:complexContent>
    <xs:restriction base="PrivateKeyProtectionType">
      <xs:sequence>
        <xs:element ref="KeyStorage"/>
        <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
      </xs:sequence>
    </xs:restriction>
  </xs:complexContent>
</xs:complexType>

<xs:complexType name="SecretKeyProtectionType">
  <xs:complexContent>
    <xs:restriction base="SecretKeyProtectionType">
      <xs:sequence>
        <xs:element ref="KeyStorage"/>
        <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
      </xs:sequence>
    </xs:restriction>
  </xs:complexContent>
</xs:complexType>

<xs:complexType name="KeyStorageType">
```

```
      <xs:complexContent>
        <xs:restriction base="KeyStorageType">
          <xs:attribute name="medium" use="required">
            <xs:simpleType>
              <xs:restriction base="mediumType">
                <xs:enumeration value="smartcard"/>
                <xs:enumeration value="MobileDevice"/>
                <xs:enumeration value="MobileAuthCard"/>
              </xs:restriction>
            </xs:simpleType>
          </xs:attribute>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="SecurityAuditType">
      <xs:complexContent>
        <xs:restriction base="SecurityAuditType">
          <xs:sequence>
            <xs:element ref="SwitchAudit"/>
            <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="IdentificationType">
      <xs:complexContent>
        <xs:restriction base="IdentificationType">
          <xs:sequence>
            <xs:element ref="PhysicalVerification"/>
            <xs:element ref="WrittenConsent"/>
            <xs:element ref="GoverningAgreements"/>
            <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
          </xs:sequence>
          <xs:attribute name="nym">
            <xs:simpleType>
              <xs:restriction base="nymType">
                <xs:enumeration value="anonymity"/>
                <xs:enumeration value="verinymity"/>
                <xs:enumeration value="pseudonymity"/>
              </xs:restriction>
            </xs:simpleType>
          </xs:attribute>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

  </xs:redefine>

</xs:schema>
```

A.8 Schéma SAML du contexte d'authentification MobileOneFactor-unreg

Cette liste contient le schéma de classe de contexte SAML pour MobileOneFactorContract non enregistré.

```
<?xml version="1.0" encoding="UTF-8"?>

<xs:schema
targetNamespace="urn:oasis:names:tc:SAML:2.0:ac:classes:MobileOneFactorUnregister
ed"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns="urn:oasis:names:tc:SAML:2.0:ac:classes:MobileOneFactorUnregistered"
  finalDefault="extension"
  blockDefault="substitution"
  version="2.0">

  <xs:redefine schemaLocation="saml-schema-authn-context-types-2.0.xsd">
```

```
    <xs:annotation>
      <xs:documentation>
        Class identifier:
urn:oasis:names:tc:SAML:2.0:ac:classes:MobileOneFactorUnregistered
        Document identifier: saml-schema-authn-context-mobileonefactor-unreg-2.0
        Location: http://docs.oasis-open.org/security/saml/v2.0/
        Revision history:
          V2.0 (March, 2005):
            New authentication_u99 context class schema for SAML V2.0.
      </xs:documentation>
    </xs:annotation>

    <xs:complexType name="AuthnContextDeclarationBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthnContextDeclarationBaseType">
          <xs:sequence>
            <xs:element ref="Identification" minOccurs="0"/>
            <xs:element ref="TechnicalProtection" minOccurs="0"/>
            <xs:element ref="OperationalProtection" minOccurs="0"/>
            <xs:element ref="AuthnMethod"/>
            <xs:element ref="GoverningAgreements" minOccurs="0"/>
            <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
          </xs:sequence>
          <xs:attribute name="ID" type="xs:ID" use="optional"/>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="AuthnMethodBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthnMethodBaseType">
          <xs:sequence>
            <xs:element ref="PrincipalAuthenticationMechanism" minOccurs="0"/>
            <xs:element ref="Authenticator"/>
            <xs:element ref="AuthenticatorTransportProtocol" minOccurs="0"/>
            <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="AuthenticatorBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthenticatorBaseType">
          <xs:sequence>
            <xs:choice>
              <xs:element ref="DigSig"/>
              <xs:element ref="ZeroKnowledge"/>
              <xs:element ref="SharedSecretChallengeResponse"/>
              <xs:element ref="SharedSecretDynamicPlaintext"/>
              <xs:element ref="AsymmetricDecryption"/>
              <xs:element ref="AsymmetricKeyAgreement"/>
            </xs:choice>
            <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="AuthenticatorTransportProtocolType">
      <xs:complexContent>
        <xs:restriction base="AuthenticatorTransportProtocolType">
          <xs:sequence>
            <xs:choice>
              <xs:element ref="SSL"/>
              <xs:element ref="MobileNetworkNoEncryption"/>
              <xs:element ref="MobileNetworkRadioEncryption"/>
              <xs:element ref="MobileNetworkEndToEndEncryption"/>
              <xs:element ref="WTLS"/>
            </xs:choice>
            <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
```

```
      </xs:sequence>
    </xs:restriction>
  </xs:complexContent>
</xs:complexType>

<xs:complexType name="OperationalProtectionType">
  <xs:complexContent>
    <xs:restriction base="OperationalProtectionType">
      <xs:sequence>
        <xs:element ref="SecurityAudit"/>
        <xs:element ref="DeactivationCallCenter"/>
        <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
      </xs:sequence>
    </xs:restriction>
  </xs:complexContent>
</xs:complexType>

<xs:complexType name="TechnicalProtectionBaseType">
  <xs:complexContent>
    <xs:restriction base="TechnicalProtectionBaseType">
      <xs:sequence>
        <xs:choice>
          <xs:element ref="PrivateKeyProtection"/>
          <xs:element ref="SecretKeyProtection"/>
        </xs:choice>
        <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
      </xs:sequence>
    </xs:restriction>
  </xs:complexContent>
</xs:complexType>

<xs:complexType name="PrivateKeyProtectionType">
  <xs:complexContent>
    <xs:restriction base="PrivateKeyProtectionType">
      <xs:sequence>
        <xs:element ref="KeyStorage"/>
        <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
      </xs:sequence>
    </xs:restriction>
  </xs:complexContent>
</xs:complexType>

<xs:complexType name="SecretKeyProtectionType">
  <xs:complexContent>
    <xs:restriction base="SecretKeyProtectionType">
      <xs:sequence>
        <xs:element ref="KeyStorage"/>
        <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
      </xs:sequence>
    </xs:restriction>
  </xs:complexContent>
</xs:complexType>

<xs:complexType name="KeyStorageType">
  <xs:complexContent>
    <xs:restriction base="KeyStorageType">
      <xs:attribute name="medium" use="required">
        <xs:simpleType>
          <xs:restriction base="mediumType">
            <xs:enumeration value="MobileDevice"/>
            <xs:enumeration value="MobileAuthCard"/>
            <xs:enumeration value="smartcard"/>
          </xs:restriction>
        </xs:simpleType>
      </xs:attribute>
    </xs:restriction>
  </xs:complexContent>
</xs:complexType>

<xs:complexType name="SecurityAuditType">
  <xs:complexContent>
```

```
        <xs:restriction base="SecurityAuditType">
          <xs:sequence>
            <xs:element ref="SwitchAudit"/>
            <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="IdentificationType">
      <xs:complexContent>
        <xs:restriction base="IdentificationType">
          <xs:sequence>
            <xs:element ref="GoverningAgreements"/>
            <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
          </xs:sequence>
          <xs:attribute name="nym">
            <xs:simpleType>
              <xs:restriction base="nymType">
                <xs:enumeration value="anonymity"/>
                <xs:enumeration value="pseudonymity"/>
              </xs:restriction>
            </xs:simpleType>
          </xs:attribute>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

  </xs:redefine>

</xs:schema>
```

A.9 Schéma SAML du contexte d'authentification MobileTwoFactor-reg

Cette liste contient le schéma de classe de contexte SAML pour MobileTwoFactorContract enregistré.

```
<?xml version="1.0" encoding="UTF-8"?>

<xs:schema
targetNamespace="urn:oasis:names:tc:SAML:2.0:ac:classes:MobileTwoFactorContract"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns="urn:oasis:names:tc:SAML:2.0:ac:classes:MobileTwoFactorContract"
  finalDefault="extension"
  blockDefault="substitution"
  version="2.0">

  <xs:redefine schemaLocation="saml-schema-authn-context-types-2.0.xsd">

    <xs:annotation>
      <xs:documentation>
        Class identifier:
urn:oasis:names:tc:SAML:2.0:ac:classes:MobileTwoFactorContract
        Document identifier: saml-schema-authn-context-mobiletwofactor-reg-2.0
        Location: http://docs.oasis-open.org/security/saml/v2.0/
        Revision history:
          V2.0 (March, 2005):
            New authentication_u99 context class schema for SAML V2.0.
      </xs:documentation>
    </xs:annotation>

    <xs:complexType name="AuthnContextDeclarationBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthnContextDeclarationBaseType">
          <xs:sequence>
            <xs:element ref="Identification" minOccurs="0"/>
            <xs:element ref="TechnicalProtection" minOccurs="0"/>
            <xs:element ref="OperationalProtection" minOccurs="0"/>
            <xs:element ref="AuthnMethod"/>
            <xs:element ref="GoverningAgreements" minOccurs="0"/>
```

```
        <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
      </xs:sequence>
      <xs:attribute name="ID" type="xs:ID" use="optional"/>
    </xs:restriction>
  </xs:complexContent>
</xs:complexType>

<xs:complexType name="AuthnMethodBaseType">
  <xs:complexContent>
    <xs:restriction base="AuthnMethodBaseType">
      <xs:sequence>
        <xs:element ref="PrincipalAuthenticationMechanism" minOccurs="0"/>
        <xs:element ref="Authenticator"/>
        <xs:element ref="AuthenticatorTransportProtocol" minOccurs="0"/>
        <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
      </xs:sequence>
    </xs:restriction>
  </xs:complexContent>
</xs:complexType>

<xs:complexType name="AuthenticatorBaseType">
  <xs:complexContent>
    <xs:restriction base="AuthenticatorBaseType">
      <xs:sequence>
        <xs:choice>
          <xs:element ref="DigSig"/>
          <xs:element ref="ZeroKnowledge"/>
          <xs:element ref="SharedSecretChallengeResponse"/>
          <xs:element ref="SharedSecretDynamicPlaintext"/>
          <xs:element ref="AsymmetricDecryption"/>
          <xs:element ref="AsymmetricKeyAgreement"/>
          <xs:element ref="ComplexAuthenticator"/>
        </xs:choice>
        <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
      </xs:sequence>
    </xs:restriction>
  </xs:complexContent>
</xs:complexType>

<xs:complexType name="ComplexAuthenticatorType">
  <xs:complexContent>
    <xs:restriction base="ComplexAuthenticatorType">
      <xs:sequence>
        <xs:choice>
          <xs:element ref="SharedSecretChallengeResponse"/>
          <xs:element ref="SharedSecretDynamicPlaintext"/>
        </xs:choice>
        <xs:element ref="Password"/>
      </xs:sequence>
    </xs:restriction>
  </xs:complexContent>
</xs:complexType>

<xs:complexType name="AuthenticatorTransportProtocolType">
  <xs:complexContent>
    <xs:restriction base="AuthenticatorTransportProtocolType">
      <xs:sequence>
        <xs:choice>
          <xs:element ref="SSL"/>
          <xs:element ref="MobileNetworkNoEncryption"/>
          <xs:element ref="MobileNetworkRadioEncryption"/>
          <xs:element ref="MobileNetworkEndToEndEncryption"/>
          <xs:element ref="WTLS"/>
        </xs:choice>
        <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
      </xs:sequence>
    </xs:restriction>
  </xs:complexContent>
</xs:complexType>

<xs:complexType name="OperationalProtectionType">
```

```
    <xs:complexContent>
      <xs:restriction base="OperationalProtectionType">
        <xs:sequence>
          <xs:element ref="SecurityAudit"/>
          <xs:element ref="DeactivationCallCenter"/>
          <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
        </xs:sequence>
      </xs:restriction>
    </xs:complexContent>
  </xs:complexType>

  <xs:complexType name="TechnicalProtectionBaseType">
    <xs:complexContent>
      <xs:restriction base="TechnicalProtectionBaseType">
        <xs:sequence>
          <xs:choice>
            <xs:element ref="PrivateKeyProtection"/>
            <xs:element ref="SecretKeyProtection"/>
          </xs:choice>
          <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
        </xs:sequence>
      </xs:restriction>
    </xs:complexContent>
  </xs:complexType>

  <xs:complexType name="PrivateKeyProtectionType">
    <xs:complexContent>
      <xs:restriction base="PrivateKeyProtectionType">
        <xs:sequence>
          <xs:element ref="KeyActivation"/>
          <xs:element ref="KeyStorage"/>
          <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
        </xs:sequence>
      </xs:restriction>
    </xs:complexContent>
  </xs:complexType>

  <xs:complexType name="SecretKeyProtectionType">
    <xs:complexContent>
      <xs:restriction base="SecretKeyProtectionType">
        <xs:sequence>
          <xs:element ref="KeyActivation"/>
          <xs:element ref="KeyStorage"/>
          <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
        </xs:sequence>
      </xs:restriction>
    </xs:complexContent>
  </xs:complexType>

  <xs:complexType name="KeyStorageType">
    <xs:complexContent>
      <xs:restriction base="KeyStorageType">
        <xs:attribute name="medium" use="required">
          <xs:simpleType>
            <xs:restriction base="mediumType">
              <xs:enumeration value="MobileDevice"/>
              <xs:enumeration value="MobileAuthCard"/>
              <xs:enumeration value="smartcard"/>
            </xs:restriction>
          </xs:simpleType>
        </xs:attribute>
      </xs:restriction>
    </xs:complexContent>
  </xs:complexType>

  <xs:complexType name="SecurityAuditType">
    <xs:complexContent>
      <xs:restriction base="SecurityAuditType">
        <xs:sequence>
          <xs:element ref="SwitchAudit"/>
          <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
```

```
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="IdentificationType">
      <xs:complexContent>
        <xs:restriction base="IdentificationType">
          <xs:sequence>
            <xs:element ref="PhysicalVerification"/>
            <xs:element ref="WrittenConsent"/>
            <xs:element ref="GoverningAgreements"/>
            <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
          </xs:sequence>
          <xs:attribute name="nym">
            <xs:simpleType>
              <xs:restriction base="nymType">
                <xs:enumeration value="anonymity"/>
                <xs:enumeration value="verinymity"/>
                <xs:enumeration value="pseudonymity"/>
              </xs:restriction>
            </xs:simpleType>
          </xs:attribute>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>
  </xs:redefine>

</xs:schema>
```

A.10 Schéma SAML du contexte d'authentification MobileTwoFactor-unreg

Cette liste contient le schéma de classe de contexte SAML pour MobileTwoFactorUnregistered.

```
<?xml version="1.0" encoding="UTF-8"?>

<xs:schema
targetNamespace="urn:oasis:names:tc:SAML:2.0:ac:classes:MobileTwoFactorUnregister
ed"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns="urn:oasis:names:tc:SAML:2.0:ac:classes:MobileTwoFactorUnregistered"
  finalDefault="extension"
  blockDefault="substitution"
  version="2.0">

  <xs:redefine schemaLocation="saml-schema-authn-context-types-2.0.xsd">

    <xs:annotation>
      <xs:documentation>
        Class identifier:
urn:oasis:names:tc:SAML:2.0:ac:classes:MobileTwoFactorUnregistered
        Document identifier: saml-schema-authn-context-mobiletwofactor-unreg-2.0
        Location: http://docs.oasis-open.org/security/saml/v2.0/
        Revision history:
          V2.0 (March, 2005):
            New authentication_u99 context class schema for SAML V2.0.
      </xs:documentation>
    </xs:annotation>

    <xs:complexType name="AuthnContextDeclarationBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthnContextDeclarationBaseType">
          <xs:sequence>
            <xs:element ref="Identification" minOccurs="0"/>
            <xs:element ref="TechnicalProtection" minOccurs="0"/>
            <xs:element ref="OperationalProtection" minOccurs="0"/>
            <xs:element ref="AuthnMethod"/>
            <xs:element ref="GoverningAgreements" minOccurs="0"/>
            <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
```

```
      </xs:sequence>
      <xs:attribute name="ID" type="xs:ID" use="optional"/>
    </xs:restriction>
  </xs:complexContent>
</xs:complexType>

<xs:complexType name="AuthnMethodBaseType">
  <xs:complexContent>
    <xs:restriction base="AuthnMethodBaseType">
      <xs:sequence>
        <xs:element ref="PrincipalAuthenticationMechanism" minOccurs="0"/>
        <xs:element ref="Authenticator"/>
        <xs:element ref="AuthenticatorTransportProtocol" minOccurs="0"/>
        <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
      </xs:sequence>
    </xs:restriction>
  </xs:complexContent>
</xs:complexType>

<xs:complexType name="AuthenticatorBaseType">
  <xs:complexContent>
    <xs:restriction base="AuthenticatorBaseType">
      <xs:sequence>
        <xs:choice>
          <xs:element ref="DigSig"/>
          <xs:element ref="ZeroKnowledge"/>
          <xs:element ref="SharedSecretChallengeResponse"/>
          <xs:element ref="SharedSecretDynamicPlaintext"/>
          <xs:element ref="AsymmetricDecryption"/>
          <xs:element ref="AsymmetricKeyAgreement"/>
          <xs:element ref="ComplexAuthenticator"/>
        </xs:choice>
        <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
      </xs:sequence>
    </xs:restriction>
  </xs:complexContent>
</xs:complexType>

<xs:complexType name="ComplexAuthenticatorType">
  <xs:complexContent>
    <xs:restriction base="ComplexAuthenticatorType">
      <xs:sequence>
        <xs:choice>
          <xs:element ref="SharedSecretChallengeResponse"/>
          <xs:element ref="SharedSecretDynamicPlaintext"/>
        </xs:choice>
        <xs:element ref="Password"/>
      </xs:sequence>
    </xs:restriction>
  </xs:complexContent>
</xs:complexType>

<xs:complexType name="AuthenticatorTransportProtocolType">
  <xs:complexContent>
    <xs:restriction base="AuthenticatorTransportProtocolType">
      <xs:sequence>
        <xs:choice>
          <xs:element ref="SSL"/>
          <xs:element ref="MobileNetworkNoEncryption"/>
          <xs:element ref="MobileNetworkRadioEncryption"/>
          <xs:element ref="MobileNetworkEndToEndEncryption"/>
          <xs:element ref="WTLS"/>
        </xs:choice>
        <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
      </xs:sequence>
    </xs:restriction>
  </xs:complexContent>
</xs:complexType>

<xs:complexType name="OperationalProtectionType">
  <xs:complexContent>
```

```
      <xs:restriction base="OperationalProtectionType">
        <xs:sequence>
          <xs:element ref="SecurityAudit"/>
          <xs:element ref="DeactivationCallCenter"/>
          <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
        </xs:sequence>
      </xs:restriction>
    </xs:complexContent>
  </xs:complexType>

  <xs:complexType name="TechnicalProtectionBaseType">
    <xs:complexContent>
      <xs:restriction base="TechnicalProtectionBaseType">
        <xs:sequence>
          <xs:choice>
            <xs:element ref="PrivateKeyProtection"/>
            <xs:element ref="SecretKeyProtection"/>
          </xs:choice>
          <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
        </xs:sequence>
      </xs:restriction>
    </xs:complexContent>
  </xs:complexType>

  <xs:complexType name="PrivateKeyProtectionType">
    <xs:complexContent>
      <xs:restriction base="PrivateKeyProtectionType">
        <xs:sequence>
          <xs:element ref="KeyActivation"/>
          <xs:element ref="KeyStorage"/>
          <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
        </xs:sequence>
      </xs:restriction>
    </xs:complexContent>
  </xs:complexType>

  <xs:complexType name="SecretKeyProtectionType">
    <xs:complexContent>
      <xs:restriction base="SecretKeyProtectionType">
        <xs:sequence>
          <xs:element ref="KeyActivation"/>
          <xs:element ref="KeyStorage"/>
          <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
        </xs:sequence>
      </xs:restriction>
    </xs:complexContent>
  </xs:complexType>

  <xs:complexType name="KeyStorageType">
    <xs:complexContent>
      <xs:restriction base="KeyStorageType">
        <xs:attribute name="medium" use="required">
          <xs:simpleType>
            <xs:restriction base="mediumType">
              <xs:enumeration value="MobileDevice"/>
              <xs:enumeration value="MobileAuthCard"/>
              <xs:enumeration value="smartcard"/>
            </xs:restriction>
          </xs:simpleType>
        </xs:attribute>
      </xs:restriction>
    </xs:complexContent>
  </xs:complexType>

  <xs:complexType name="SecurityAuditType">
    <xs:complexContent>
      <xs:restriction base="SecurityAuditType">
        <xs:sequence>
          <xs:element ref="SwitchAudit"/>
          <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
        </xs:sequence>
```

```
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="IdentificationType">
      <xs:complexContent>
        <xs:restriction base="IdentificationType">
          <xs:sequence>
            <xs:element ref="GoverningAgreements"/>
            <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
          </xs:sequence>
          <xs:attribute name="nym">
            <xs:simpleType>
              <xs:restriction base="nymType">
                <xs:enumeration value="anonymity"/>
                <xs:enumeration value="pseudonymity"/>
              </xs:restriction>
            </xs:simpleType>
          </xs:attribute>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

  </xs:redefine>

</xs:schema>
```

A.11 Schéma SAML du contexte d'authentification NomadTelephony

Cette liste contient le schéma SAML d'authentification de NomadTelephony. Téléphonie nomade indique que le principal est "en itinérance" (utilisant peut-être une carte téléphonique) et s'authentifie au moyen d'un numéro de ligne téléphonique, d'un suffixe d'utilisateur, et d'un élément mot de passe.

```
<?xml version="1.0" encoding="UTF-8"?>

<xs:schema
targetNamespace="urn:oasis:names:tc:SAML:2.0:ac:classes:NomadTelephony"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns="urn:oasis:names:tc:SAML:2.0:ac:classes:NomadTelephony"
  finalDefault="extension"
  blockDefault="substitution"
  version="2.0">

  <xs:redefine schemaLocation="saml-schema-authn-context-types-2.0.xsd">

    <xs:annotation>
      <xs:documentation>
        Class identifier: urn:oasis:names:tc:SAML:2.0:ac:classes:NomadTelephony
        Document identifier: saml-schema-authn-context-nomad-telephony-2.0
        Location: http://docs.oasis-open.org/security/saml/v2.0/
        Revision history:
          V2.0 (March, 2005):
            New authentication_u99 context class schema for SAML V2.0.
      </xs:documentation>
    </xs:annotation>

    <xs:complexType name="AuthnContextDeclarationBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthnContextDeclarationBaseType">
          <xs:sequence>
            <xs:element ref="Identification" minOccurs="0"/>
            <xs:element ref="TechnicalProtection" minOccurs="0"/>
            <xs:element ref="OperationalProtection" minOccurs="0"/>
            <xs:element ref="AuthnMethod"/>
            <xs:element ref="GoverningAgreements" minOccurs="0"/>
            <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
          </xs:sequence>
          <xs:attribute name="ID" type="xs:ID" use="optional"/>
        </xs:restriction>
```

```
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="AuthnMethodBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthnMethodBaseType">
          <xs:sequence>
            <xs:element ref="PrincipalAuthenticationMechanism" minOccurs="0"/>
            <xs:element ref="Authenticator"/>
            <xs:element ref="AuthenticatorTransportProtocol"/>
            <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="AuthenticatorBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthenticatorBaseType">
          <xs:sequence>
            <xs:element ref="Password"/>
            <xs:element ref="SubscriberLineNumber"/>
            <xs:element ref="UserSuffix"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="AuthenticatorTransportProtocolType">
      <xs:complexContent>
        <xs:restriction base="AuthenticatorTransportProtocolType">
          <xs:sequence>
            <xs:choice>
              <xs:element ref="PSTN"/>
              <xs:element ref="ISDN"/>
              <xs:element ref="ADSL"/>
            </xs:choice>
            <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

  </xs:redefine>

</xs:schema>
```

A.12 Schéma SAML du contexte d'authentification PersonalizedTelephony

Cette liste fournit le schéma SAML d'authentification pour la téléphonie personnelle.

```
<?xml version="1.0" encoding="UTF-8"?>

<xs:schema
targetNamespace="urn:oasis:names:tc:SAML:2.0:ac:classes:PersonalizedTelephony"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns="urn:oasis:names:tc:SAML:2.0:ac:classes:PersonalizedTelephony"
  finalDefault="extension"
  blockDefault="substitution"
  version="2.0">

  <xs:redefine schemaLocation="saml-schema-authn-context-types-2.0.xsd">

    <xs:annotation>
      <xs:documentation>
        Class identifier:
urn:oasis:names:tc:SAML:2.0:ac:classes:PersonalizedTelephony
        Document identifier: saml-schema-authn-context-personal-telephony-2.0
        Location: http://docs.oasis-open.org/security/saml/v2.0/
```

```
        Revision history:
          V2.0 (March, 2005):
            New authentication_u99 context class schema for SAML V2.0.
      </xs:documentation>
    </xs:annotation>

    <xs:complexType name="AuthnContextDeclarationBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthnContextDeclarationBaseType">
          <xs:sequence>
            <xs:element ref="Identification" minOccurs="0"/>
            <xs:element ref="TechnicalProtection" minOccurs="0"/>
            <xs:element ref="OperationalProtection" minOccurs="0"/>
            <xs:element ref="AuthnMethod"/>
            <xs:element ref="GoverningAgreements" minOccurs="0"/>
            <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
          </xs:sequence>
          <xs:attribute name="ID" type="xs:ID" use="optional"/>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="AuthnMethodBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthnMethodBaseType">
          <xs:sequence>
            <xs:element ref="PrincipalAuthenticationMechanism" minOccurs="0"/>
            <xs:element ref="Authenticator"/>
            <xs:element ref="AuthenticatorTransportProtocol"/>
            <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="AuthenticatorBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthenticatorBaseType">
          <xs:sequence>
            <xs:element ref="SubscriberLineNumber"/>
            <xs:element ref="UserSuffix"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="AuthenticatorTransportProtocolType">
      <xs:complexContent>
        <xs:restriction base="AuthenticatorTransportProtocolType">
          <xs:sequence>
            <xs:choice>
              <xs:element ref="PSTN"/>
              <xs:element ref="ISDN"/>
              <xs:element ref="ADSL"/>
            </xs:choice>
            <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

  </xs:redefine>

</xs:schema>
```

A.13 Schéma SAML du contexte d'authentification PGP

Cette liste fournit le schéma SAML d'authentification pour PGP.

```
<?xml version="1.0" encoding="UTF-8"?>

<xs:schema targetNamespace="urn:oasis:names:tc:SAML:2.0:ac:classes:PGP"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns="urn:oasis:names:tc:SAML:2.0:ac:classes:PGP"
  finalDefault="extension"
  blockDefault="substitution"
  version="2.0">

  <xs:redefine schemaLocation="saml-schema-authn-context-types-2.0.xsd">

    <xs:annotation>
      <xs:documentation>
        Class identifier: urn:oasis:names:tc:SAML:2.0:ac:classes:PGP
        Document identifier: saml-schema-authn-context-pgp-2.0
        Location: http://docs.oasis-open.org/security/saml/v2.0/
        Revision history:
          V2.0 (March, 2005):
            New authentication_u99 context class schema for SAML V2.0.
      </xs:documentation>
    </xs:annotation>

    <xs:complexType name="AuthnContextDeclarationBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthnContextDeclarationBaseType">
          <xs:sequence>
            <xs:element ref="Identification" minOccurs="0"/>
            <xs:element ref="TechnicalProtection" minOccurs="0"/>
            <xs:element ref="OperationalProtection" minOccurs="0"/>
            <xs:element ref="AuthnMethod"/>
            <xs:element ref="GoverningAgreements" minOccurs="0"/>
            <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
          </xs:sequence>
          <xs:attribute name="ID" type="xs:ID" use="optional"/>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="AuthnMethodBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthnMethodBaseType">
          <xs:sequence>
            <xs:element ref="PrincipalAuthenticationMechanism"/>
            <xs:element ref="Authenticator"/>
            <xs:element ref="AuthenticatorTransportProtocol" minOccurs="0"/>
            <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="PrincipalAuthenticationMechanismType">
      <xs:complexContent>
        <xs:restriction base="PrincipalAuthenticationMechanismType">
          <xs:sequence>
            <xs:element ref="RestrictedPassword"/>
          </xs:sequence>
          <xs:attribute name="preauth" type="xs:integer" use="optional"/>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="AuthenticatorBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthenticatorBaseType">
          <xs:sequence>
            <xs:element ref="DigSig"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>
```

```
    <xs:complexType name="PublicKeyType">
      <xs:complexContent>
        <xs:restriction base="PublicKeyType">
          <xs:attribute name="keyValidation"
fixed="urn:oasis:names:tc:SAML:2.0:ac:classes:PGP"/>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

  </xs:redefine>
</xs:schema>
```

A.14 Schéma SAML du contexte d'authentification PPT

Cette liste contient le schéma SAML d'authentification pour le transport protégé par mot de passe (PPT, *password protected transport*).

```
<?xml version="1.0" encoding="UTF-8"?>

<xs:schema
targetNamespace="urn:oasis:names:tc:SAML:2.0:ac:classes:PasswordProtectedTranspor
t"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns="urn:oasis:names:tc:SAML:2.0:ac:classes:PasswordProtectedTransport"
  finalDefault="extension"
  blockDefault="substitution"
  version="2.0">

  <xs:redefine schemaLocation="saml-schema-authn-context-types-2.0.xsd">

    <xs:annotation>
      <xs:documentation>
        Class identifier:
urn:oasis:names:tc:SAML:2.0:ac:classes:PasswordProtectedTransport
        Document identifier: saml-schema-authn-context-ppt-2.0
        Location: http://docs.oasis-open.org/security/saml/v2.0/
        Revision history:
          V2.0 (March, 2005):
            New authentication_u99 context class schema for SAML V2.0.
      </xs:documentation>
    </xs:annotation>

    <xs:complexType name="AuthnContextDeclarationBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthnContextDeclarationBaseType">
          <xs:sequence>
            <xs:element ref="Identification" minOccurs="0"/>
            <xs:element ref="TechnicalProtection" minOccurs="0"/>
            <xs:element ref="OperationalProtection" minOccurs="0"/>
            <xs:element ref="AuthnMethod"/>
            <xs:element ref="GoverningAgreements" minOccurs="0"/>
            <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
          </xs:sequence>
          <xs:attribute name="ID" type="xs:ID" use="optional"/>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="AuthnMethodBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthnMethodBaseType">
          <xs:sequence>
            <xs:element ref="PrincipalAuthenticationMechanism" minOccurs="0"/>
            <xs:element ref="Authenticator"/>
            <xs:element ref="AuthenticatorTransportProtocol"/>
            <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:restriction>
```

```
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="AuthenticatorBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthenticatorBaseType">
          <xs:sequence>
            <xs:element ref="RestrictedPassword"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="AuthenticatorTransportProtocolType">
      <xs:complexContent>
        <xs:restriction base="AuthenticatorTransportProtocolType">
          <xs:sequence>
            <xs:choice>
              <xs:element ref="SSL"/>
              <xs:element ref="MobileNetworkRadioEncryption"/>
              <xs:element ref="MobileNetworkEndToEndEncryption"/>
              <xs:element ref="WTLS"/>
              <xs:element ref="IPSec"/>
            </xs:choice>
            <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

  </xs:redefine>

</xs:schema>
```

A.15 Schéma SAML du contexte d'authentification Password

Cette liste contient le schéma SAML de contexte d'authentification Password.

```
<?xml version="1.0" encoding="UTF-8"?>

<xs:schema targetNamespace="urn:oasis:names:tc:SAML:2.0:ac:classes:Password"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns="urn:oasis:names:tc:SAML:2.0:ac:classes:Password"
  finalDefault="extension"
  blockDefault="substitution"
  version="2.0">

  <xs:redefine schemaLocation="saml-schema-authn-context-types-2.0.xsd">

    <xs:annotation>
      <xs:documentation>
        Class identifier: urn:oasis:names:tc:SAML:2.0:ac:classes:Password
        Document identifier: saml-schema-authn-context-pword-2.0
        Location: http://docs.oasis-open.org/security/saml/v2.0/
        Revision history:
          V2.0 (March, 2005):
            New authentication_u99 context class schema for SAML V2.0.
      </xs:documentation>
    </xs:annotation>

    <xs:complexType name="AuthnContextDeclarationBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthnContextDeclarationBaseType">
          <xs:sequence>
            <xs:element ref="Identification" minOccurs="0"/>
            <xs:element ref="TechnicalProtection" minOccurs="0"/>
            <xs:element ref="OperationalProtection" minOccurs="0"/>
            <xs:element ref="AuthnMethod"/>
            <xs:element ref="GoverningAgreements" minOccurs="0"/>
```

```
            <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
          </xs:sequence>
          <xs:attribute name="ID" type="xs:ID" use="optional"/>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="AuthnMethodBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthnMethodBaseType">
          <xs:sequence>
            <xs:element ref="PrincipalAuthenticationMechanism" minOccurs="0"/>
            <xs:element ref="Authenticator"/>
            <xs:element ref="AuthenticatorTransportProtocol" minOccurs="0"/>
            <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="AuthenticatorBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthenticatorBaseType">
          <xs:sequence>
            <xs:element ref="RestrictedPassword"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

  </xs:redefine>

</xs:schema>
```

A.16 Schéma SAML du contexte d'authentification PreviousSession

Cette liste contient le schéma SAML de contexte d'authentification pour PreviousSession. La classe PreviousSession est applicable lorsqu'un principal s'est authentifié auprès d'une autorité d'authentification à un moment dans le passé en utilisant un des contextes d'authentification acceptés par cette autorité d'authentification.

```
<?xml version="1.0" encoding="UTF-8"?>

<xs:schema
targetNamespace="urn:oasis:names:tc:SAML:2.0:ac:classes:PreviousSession"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns="urn:oasis:names:tc:SAML:2.0:ac:classes:PreviousSession"
  finalDefault="extension"
  blockDefault="substitution"
  version="2.0">

  <xs:redefine schemaLocation="saml-schema-authn-context-types-2.0.xsd">

    <xs:annotation>
      <xs:documentation>
        Class identifier: urn:oasis:names:tc:SAML:2.0:ac:classes:PreviousSession
        Document identifier: saml-schema-authn-context-session-2.0
        Location: http://docs.oasis-open.org/security/saml/v2.0/
        Revision history:
          V2.0 (March, 2005):
            New authentication_u99 context class schema for SAML V2.0.
      </xs:documentation>
    </xs:annotation>

    <xs:complexType name="AuthnContextDeclarationBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthnContextDeclarationBaseType">
          <xs:sequence>
            <xs:element ref="Identification" minOccurs="0"/>
            <xs:element ref="TechnicalProtection" minOccurs="0"/>
```

```
            <xs:element ref="OperationalProtection" minOccurs="0"/>
            <xs:element ref="AuthnMethod"/>
            <xs:element ref="GoverningAgreements" minOccurs="0"/>
            <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
          </xs:sequence>
          <xs:attribute name="ID" type="xs:ID" use="optional"/>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="AuthnMethodBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthnMethodBaseType">
          <xs:sequence>
            <xs:element ref="PrincipalAuthenticationMechanism" minOccurs="0"/>
            <xs:element ref="Authenticator"/>
            <xs:element ref="AuthenticatorTransportProtocol" minOccurs="0"/>
            <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="AuthenticatorBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthenticatorBaseType">
          <xs:sequence>
            <xs:element ref="PreviousSession"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

  </xs:redefine>

</xs:schema>
```

A.17 Schéma SAML du contexte d'authentification Smartcard

Schéma SAML de contexte d'authentification pour carte à mémoire.

```
<?xml version="1.0" encoding="UTF-8"?>

<xs:schema targetNamespace="urn:oasis:names:tc:SAML:2.0:ac:classes:Smartcard"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns="urn:oasis:names:tc:SAML:2.0:ac:classes:Smartcard"
  finalDefault="extension"
  blockDefault="substitution"
  version="2.0">

  <xs:redefine schemaLocation="saml-schema-authn-context-types-2.0.xsd">

    <xs:annotation>
      <xs:documentation>
        Class identifier: urn:oasis:names:tc:SAML:2.0:ac:classes:Smartcard
        Document identifier: saml-schema-authn-context-smartcard-2.0
        Location: http://docs.oasis-open.org/security/saml/v2.0/
        Revision history:
          V2.0 (March, 2005):
            New authentication_u99 context class schema for SAML V2.0.
      </xs:documentation>
    </xs:annotation>

    <xs:complexType name="AuthnContextDeclarationBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthnContextDeclarationBaseType">
          <xs:sequence>
            <xs:element ref="Identification" minOccurs="0"/>
            <xs:element ref="TechnicalProtection" minOccurs="0"/>
```

```
            <xs:element ref="OperationalProtection" minOccurs="0"/>
            <xs:element ref="AuthnMethod"/>
            <xs:element ref="GoverningAgreements" minOccurs="0"/>
            <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
          </xs:sequence>
          <xs:attribute name="ID" type="xs:ID" use="optional"/>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="AuthnMethodBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthnMethodBaseType">
          <xs:sequence>
            <xs:element ref="PrincipalAuthenticationMechanism"/>
            <xs:element ref="Authenticator"/>
            <xs:element ref="AuthenticatorTransportProtocol" minOccurs="0"/>
            <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="PrincipalAuthenticationMechanismType">
      <xs:complexContent>
        <xs:restriction base="PrincipalAuthenticationMechanismType">
          <xs:sequence>
            <xs:element ref="Smartcard"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

  </xs:redefine>

</xs:schema>
```

A.18 Schéma SAML du contexte d'authentification SmartcardPKI

Schéma SAML de contexte d'authentification pour carte à mémoire PKI.

```
<?xml version="1.0" encoding="UTF-8"?>

<xs:schema targetNamespace="urn:oasis:names:tc:SAML:2.0:ac:classes:SmartcardPKI"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns="urn:oasis:names:tc:SAML:2.0:ac:classes:SmartcardPKI"
  finalDefault="extension"
  blockDefault="substitution"
  version="2.0">

  <xs:redefine schemaLocation="saml-schema-authn-context-types-2.0.xsd">

    <xs:annotation>
      <xs:documentation>
        Class identifier: urn:oasis:names:tc:SAML:2.0:ac:classes:SmartcardPKI
        Document identifier: saml-schema-authn-context-smartcardpki-2.0
        Location: http://docs.oasis-open.org/security/saml/v2.0/
        Revision history:
          V2.0 (March, 2005):
            New authentication_u99 context class schema for SAML V2.0.
      </xs:documentation>
    </xs:annotation>

    <xs:complexType name="AuthnContextDeclarationBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthnContextDeclarationBaseType">
          <xs:sequence>
            <xs:element ref="Identification" minOccurs="0"/>
            <xs:element ref="TechnicalProtection"/>
```

```
        <xs:element ref="OperationalProtection" minOccurs="0"/>
        <xs:element ref="AuthnMethod"/>
        <xs:element ref="GoverningAgreements" minOccurs="0"/>
        <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
      </xs:sequence>
      <xs:attribute name="ID" type="xs:ID" use="optional"/>
    </xs:restriction>
  </xs:complexContent>
</xs:complexType>

<xs:complexType name="AuthnMethodBaseType">
  <xs:complexContent>
    <xs:restriction base="AuthnMethodBaseType">
      <xs:sequence>
        <xs:element ref="PrincipalAuthenticationMechanism"/>
        <xs:element ref="Authenticator"/>
        <xs:element ref="AuthenticatorTransportProtocol" minOccurs="0"/>
        <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
      </xs:sequence>
    </xs:restriction>
  </xs:complexContent>
</xs:complexType>

<xs:complexType name="TechnicalProtectionBaseType">
  <xs:complexContent>
    <xs:restriction base="TechnicalProtectionBaseType">
      <xs:sequence>
        <xs:choice>
          <xs:element ref="PrivateKeyProtection"/>
        </xs:choice>
      </xs:sequence>
    </xs:restriction>
  </xs:complexContent>
</xs:complexType>

<xs:complexType name="PrincipalAuthenticationMechanismType">
  <xs:complexContent>
    <xs:restriction base="PrincipalAuthenticationMechanismType">
      <xs:sequence>
        <xs:element ref="Smartcard"/>
        <xs:element ref="ActivationPin"/>
        <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
      </xs:sequence>
    </xs:restriction>
  </xs:complexContent>
</xs:complexType>

<xs:complexType name="AuthenticatorBaseType">
  <xs:complexContent>
    <xs:restriction base="AuthenticatorBaseType">
      <xs:sequence>
        <xs:choice>
          <xs:element ref="DigSig"/>
          <xs:element ref="AsymmetricDecryption"/>
          <xs:element ref="AsymmetricKeyAgreement"/>
        </xs:choice>
        <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
      </xs:sequence>
    </xs:restriction>
  </xs:complexContent>
</xs:complexType>

<xs:complexType name="PrivateKeyProtectionType">
  <xs:complexContent>
    <xs:restriction base="PrivateKeyProtectionType">
      <xs:sequence>
        <xs:element ref="KeyActivation"/>
        <xs:element ref="KeyStorage"/>
        <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
      </xs:sequence>
    </xs:restriction>
```

```
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="KeyActivationType">
      <xs:complexContent>
        <xs:restriction base="KeyActivationType">
          <xs:sequence>
            <xs:element ref="ActivationPin"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="KeyStorageType">
      <xs:complexContent>
        <xs:restriction base="KeyStorageType">
          <xs:attribute name="medium" use="required">
            <xs:simpleType>
              <xs:restriction base="mediumType">
                <xs:enumeration value="smartcard"/>
              </xs:restriction>
            </xs:simpleType>
          </xs:attribute>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

  </xs:redefine>

</xs:schema>
```

A.19 Schéma SAML du contexte d'authentification SoftwarePKI

Schéma SAML du contexte d'authentification de logiciel PKI.

```
<?xml version="1.0" encoding="UTF-8"?>

<xs:schema targetNamespace="urn:oasis:names:tc:SAML:2.0:ac:classes:SoftwarePKI"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns="urn:oasis:names:tc:SAML:2.0:ac:classes:SoftwarePKI"
  finalDefault="extension"
  blockDefault="substitution"
  version="2.0">

  <xs:redefine schemaLocation="saml-schema-authn-context-types-2.0.xsd">

    <xs:annotation>
      <xs:documentation>
        Class identifier: urn:oasis:names:tc:SAML:2.0:ac:classes:SoftwarePKI
        Document identifier: saml-schema-authn-context-softwarepki-2.0
        Location: http://docs.oasis-open.org/security/saml/v2.0/
        Revision history:
          V2.0 (March, 2005):
            New authentication_u99 context class schema for SAML V2.0.
      </xs:documentation>
    </xs:annotation>

    <xs:complexType name="AuthnContextDeclarationBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthnContextDeclarationBaseType">
          <xs:sequence>
            <xs:element ref="Identification" minOccurs="0"/>
            <xs:element ref="TechnicalProtection"/>
            <xs:element ref="OperationalProtection" minOccurs="0"/>
            <xs:element ref="AuthnMethod"/>
            <xs:element ref="GoverningAgreements" minOccurs="0"/>
            <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
          </xs:sequence>
          <xs:attribute name="ID" type="xs:ID" use="optional"/>
```

```
      </xs:restriction>
    </xs:complexContent>
  </xs:complexType>

  <xs:complexType name="AuthnMethodBaseType">
    <xs:complexContent>
      <xs:restriction base="AuthnMethodBaseType">
        <xs:sequence>
          <xs:element ref="PrincipalAuthenticationMechanism"/>
          <xs:element ref="Authenticator"/>
          <xs:element ref="AuthenticatorTransportProtocol" minOccurs="0"/>
          <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
        </xs:sequence>
      </xs:restriction>
    </xs:complexContent>
  </xs:complexType>

  <xs:complexType name="TechnicalProtectionBaseType">
    <xs:complexContent>
      <xs:restriction base="TechnicalProtectionBaseType">
        <xs:sequence>
          <xs:choice>
            <xs:element ref="PrivateKeyProtection"/>
          </xs:choice>
        </xs:sequence>
      </xs:restriction>
    </xs:complexContent>
  </xs:complexType>

  <xs:complexType name="PrincipalAuthenticationMechanismType">
    <xs:complexContent>
      <xs:restriction base="PrincipalAuthenticationMechanismType">
        <xs:sequence>
          <xs:element ref="ActivationPin"/>
          <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
        </xs:sequence>
      </xs:restriction>
    </xs:complexContent>
  </xs:complexType>

  <xs:complexType name="AuthenticatorBaseType">
    <xs:complexContent>
      <xs:restriction base="AuthenticatorBaseType">
        <xs:sequence>
          <xs:choice>
            <xs:element ref="DigSig"/>
            <xs:element ref="AsymmetricDecryption"/>
            <xs:element ref="AsymmetricKeyAgreement"/>
          </xs:choice>
          <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
        </xs:sequence>
      </xs:restriction>
    </xs:complexContent>
  </xs:complexType>

  <xs:complexType name="PrivateKeyProtectionType">
    <xs:complexContent>
      <xs:restriction base="PrivateKeyProtectionType">
        <xs:sequence>
          <xs:element ref="KeyActivation"/>
          <xs:element ref="KeyStorage"/>
          <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
        </xs:sequence>
      </xs:restriction>
    </xs:complexContent>
  </xs:complexType>

  <xs:complexType name="KeyActivationType">
    <xs:complexContent>
      <xs:restriction base="KeyActivationType">
        <xs:sequence>
```

```
            <xs:element ref="ActivationPin"/>
            <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="KeyStorageType">
      <xs:complexContent>
        <xs:restriction base="KeyStorageType">
          <xs:attribute name="medium" use="required">
            <xs:simpleType>
              <xs:restriction base="mediumType">
                <xs:enumeration value="memory"/>
              </xs:restriction>
            </xs:simpleType>
          </xs:attribute>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

  </xs:redefine>

</xs:schema>
```

A.20 Schéma SAML du contexte d'authentification SPKI

Schéma SAML du contexte d'authentification de clés publiques. La classe de contexte SPKI indique que le principal s'est authentifié au moyen d'une signature numérique où la clé a été validée via une infrastructure SPKI.

```
<?xml version="1.0" encoding="UTF-8"?>

<xs:schema targetNamespace="urn:oasis:names:tc:SAML:2.0:ac:classes:SPKI"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns="urn:oasis:names:tc:SAML:2.0:ac:classes:SPKI"
  finalDefault="extension"
  blockDefault="substitution"
  version="2.0">

  <xs:redefine schemaLocation="saml-schema-authn-context-types-2.0.xsd">

    <xs:annotation>
      <xs:documentation>
        Class identifier: urn:oasis:names:tc:SAML:2.0:ac:classes:SPKI
        Document identifier: saml-schema-authn-context-spki-2.0
        Location: http://docs.oasis-open.org/security/saml/v2.0/
        Revision history:
          V2.0 (March, 2005):
            New authentication_u99 context class schema for SAML V2.0.
      </xs:documentation>
    </xs:annotation>

    <xs:complexType name="AuthnContextDeclarationBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthnContextDeclarationBaseType">
          <xs:sequence>
            <xs:element ref="Identification" minOccurs="0"/>
            <xs:element ref="TechnicalProtection" minOccurs="0"/>
            <xs:element ref="OperationalProtection" minOccurs="0"/>
            <xs:element ref="AuthnMethod"/>
            <xs:element ref="GoverningAgreements" minOccurs="0"/>
            <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
          </xs:sequence>
          <xs:attribute name="ID" type="xs:ID" use="optional"/>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="AuthnMethodBaseType">
```

```
      <xs:complexContent>
        <xs:restriction base="AuthnMethodBaseType">
          <xs:sequence>
            <xs:element ref="PrincipalAuthenticationMechanism"/>
            <xs:element ref="Authenticator"/>
            <xs:element ref="AuthenticatorTransportProtocol" minOccurs="0"/>
            <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="PrincipalAuthenticationMechanismType">
      <xs:complexContent>
        <xs:restriction base="PrincipalAuthenticationMechanismType">
          <xs:sequence>
            <xs:element ref="RestrictedPassword"/>
          </xs:sequence>
          <xs:attribute name="preauth" type="xs:integer" use="optional"/>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="AuthenticatorBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthenticatorBaseType">
          <xs:sequence>
            <xs:element ref="DigSig"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="PublicKeyType">
      <xs:complexContent>
        <xs:restriction base="PublicKeyType">
          <xs:attribute name="keyValidation"
fixed="urn:oasis:names:tc:SAML:2.0:ac:classes:SPKI"/>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

  </xs:redefine>

</xs:schema>
```

A.21 Schéma SAML du contexte d'authentification SRP

Schéma SAML du contexte d'authentification avec mot de passe distant sécurisé (SRP) [voir la RFC 2945 de l'IETF].

```
<?xml version="1.0" encoding="UTF-8"?>

<xs:schema
targetNamespace="urn:oasis:names:tc:SAML:2.0:ac:classes:SecureRemotePassword"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns="urn:oasis:names:tc:SAML:2.0:ac:classes:SecureRemotePassword"
  finalDefault="extension"
  blockDefault="substitution"
  version="2.0">

  <xs:redefine schemaLocation="saml-schema-authn-context-types-2.0.xsd">

    <xs:annotation>
      <xs:documentation>
        Class identifier:
urn:oasis:names:tc:SAML:2.0:ac:classes:SecureRemotePassword
        Document identifier: saml-schema-authn-context-srp-2.0
        Location: http://docs.oasis-open.org/security/saml/v2.0/
        Revision history:
```

```
          V2.0 (March, 2005):
            New authentication_u99 context class schema for SAML V2.0.
      </xs:documentation>
    </xs:annotation>

    <xs:complexType name="AuthnContextDeclarationBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthnContextDeclarationBaseType">
          <xs:sequence>
            <xs:element ref="Identification" minOccurs="0"/>
            <xs:element ref="TechnicalProtection" minOccurs="0"/>
            <xs:element ref="OperationalProtection" minOccurs="0"/>
            <xs:element ref="AuthnMethod"/>
            <xs:element ref="GoverningAgreements" minOccurs="0"/>
            <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
          </xs:sequence>
          <xs:attribute name="ID" type="xs:ID" use="optional"/>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="AuthnMethodBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthnMethodBaseType">
          <xs:sequence>
            <xs:element ref="PrincipalAuthenticationMechanism"/>
            <xs:element ref="Authenticator"/>
            <xs:element ref="AuthenticatorTransportProtocol" minOccurs="0"/>
            <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="PrincipalAuthenticationMechanismType">
      <xs:complexContent>
        <xs:restriction base="PrincipalAuthenticationMechanismType">
          <xs:sequence>
            <xs:element ref="RestrictedPassword"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="AuthenticatorBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthenticatorBaseType">
          <xs:sequence>
            <xs:element ref="SharedSecretChallengeResponse"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="SharedSecretChallengeResponseType">
      <xs:complexContent>
        <xs:restriction base="SharedSecretChallengeResponseType">
          <xs:attribute name="method" type="xs:anyURI"
fixed="urn:ietf:rfc:2945"/>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

  </xs:redefine>

</xs:schema>
```

A.22 Schéma SAML du contexte d'authentification Telephony

Schéma SAML du contexte d'authentification Telephony. Il est utilisé lorsque le principal s'authentifie via la fourniture d'un numéro de téléphone fixe, transporté via un protocole téléphonique.

```
<?xml version="1.0" encoding="UTF-8"?>

<xs:schema targetNamespace="urn:oasis:names:tc:SAML:2.0:ac:classes:Telephony"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns="urn:oasis:names:tc:SAML:2.0:ac:classes:Telephony"
  finalDefault="extension"
  blockDefault="substitution"
  version="2.0">

  <xs:redefine schemaLocation="saml-schema-authn-context-types-2.0.xsd">

    <xs:annotation>
      <xs:documentation>
        Class identifier: urn:oasis:names:tc:SAML:2.0:ac:classes:Telephony
        Document identifier: saml-schema-authn-context-telephony-2.0
        Location: http://docs.oasis-open.org/security/saml/v2.0/
        Revision history:
          V2.0 (March, 2005):
            New authentication_u99 context class schema for SAML V2.0.
      </xs:documentation>
    </xs:annotation>

    <xs:complexType name="AuthnContextDeclarationBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthnContextDeclarationBaseType">
          <xs:sequence>
            <xs:element ref="Identification" minOccurs="0"/>
            <xs:element ref="TechnicalProtection" minOccurs="0"/>
            <xs:element ref="OperationalProtection" minOccurs="0"/>
            <xs:element ref="AuthnMethod"/>
            <xs:element ref="GoverningAgreements" minOccurs="0"/>
            <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
          </xs:sequence>
          <xs:attribute name="ID" type="xs:ID" use="optional"/>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="AuthnMethodBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthnMethodBaseType">
          <xs:sequence>
            <xs:element ref="PrincipalAuthenticationMechanism" minOccurs="0"/>
            <xs:element ref="Authenticator"/>
            <xs:element ref="AuthenticatorTransportProtocol"/>
            <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="AuthenticatorBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthenticatorBaseType">
          <xs:sequence>
            <xs:element ref="SubscriberLineNumber"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="AuthenticatorTransportProtocolType">
      <xs:complexContent>
        <xs:restriction base="AuthenticatorTransportProtocolType">
          <xs:sequence>
```

```
            <xs:choice>
              <xs:element ref="PSTN"/>
              <xs:element ref="ISDN"/>
              <xs:element ref="ADSL"/>
            </xs:choice>
            <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

  </xs:redefine>
</xs:schema>
```

A.23 Schéma SAML du contexte d'authentification TimeSync

Schéma SAML du contexte d'authentification TimeSynchToken. TimeSyncToken s'applique lorsqu'un principal s'authentifie au moyen d'un jeton de synchronisation.

```
<?xml version="1.0" encoding="UTF-8"?>

<xs:schema targetNamespace="urn:oasis:names:tc:SAML:2.0:ac:classes:TimeSyncToken"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns="urn:oasis:names:tc:SAML:2.0:ac:classes:TimeSyncToken"
  finalDefault="extension"
  blockDefault="substitution"
  version="2.0">

  <xs:redefine schemaLocation="saml-schema-authn-context-types-2.0.xsd">

    <xs:annotation>
      <xs:documentation>
        Class identifier: urn:oasis:names:tc:SAML:2.0:ac:classes:TimeSyncToken
        Document identifier: saml-schema-authn-context-timesync-2.0
        Location: http://docs.oasis-open.org/security/saml/v2.0/
        Revision history:
          V2.0 (March, 2005):
            New authentication_u99 context class schema for SAML V2.0.
      </xs:documentation>
    </xs:annotation>

    <xs:complexType name="AuthnContextDeclarationBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthnContextDeclarationBaseType">
          <xs:sequence>
            <xs:element ref="Identification" minOccurs="0"/>
            <xs:element ref="TechnicalProtection" minOccurs="0"/>
            <xs:element ref="OperationalProtection" minOccurs="0"/>
            <xs:element ref="AuthnMethod"/>
            <xs:element ref="GoverningAgreements" minOccurs="0"/>
            <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
          </xs:sequence>
          <xs:attribute name="ID" type="xs:ID" use="optional"/>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="AuthnMethodBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthnMethodBaseType">
          <xs:sequence>
            <xs:element ref="PrincipalAuthenticationMechanism" minOccurs="0"/>
            <xs:element ref="Authenticator"/>
            <xs:element ref="AuthenticatorTransportProtocol" minOccurs="0"/>
            <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>
```

```
    <xs:complexType name="PrincipalAuthenticationMechanismType">
      <xs:complexContent>
        <xs:restriction base="PrincipalAuthenticationMechanismType">
          <xs:sequence>
            <xs:element ref="Token"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="TokenType">
      <xs:complexContent>
        <xs:restriction base="TokenType">
          <xs:sequence>
            <xs:element ref="TimeSyncToken"/>
            <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="TimeSyncTokenType">
      <xs:complexContent>
        <xs:restriction base="TimeSyncTokenType">
          <xs:attribute name="DeviceType" use="required">
            <xs:simpleType>
              <xs:restriction base="DeviceTypeType">
                <xs:enumeration value="hardware"/>
              </xs:restriction>
            </xs:simpleType>
          </xs:attribute>

          <xs:attribute name="SeedLength" use="required">
            <xs:simpleType>
              <xs:restriction base="xs:integer">
                <xs:minInclusive value="64"/>
              </xs:restriction>
            </xs:simpleType>
          </xs:attribute>

          <xs:attribute name="DeviceInHand" use="required">
            <xs:simpleType>
              <xs:restriction base="booleanType">
                <xs:enumeration value="true"/>
              </xs:restriction>
            </xs:simpleType>
          </xs:attribute>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

  </xs:redefine>

</xs:schema>
```

A.24 Schéma SAML du contexte d'authentification types

Schéma SAML du contexte d'authentification types.

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  elementFormDefault="qualified"
  version="2.0">

  <xs:annotation>
    <xs:documentation>
      Document identifier: saml-schema-authn-context-types-2.0
```

```
      Location: http://docs.oasis-open.org/security/saml/v2.0/
      Revision history:
          V2.0 (March, 2005):
          New core authentication context schema types for SAML V2.0.
    </xs:documentation>
  </xs:annotation>

  <xs:element name="AuthenticationContextDeclaration"
type="AuthnContextDeclarationBaseType">
    <xs:annotation>
      <xs:documentation>
        A particular assertion_u111 on an identity
        provider's part with respect to the authentication
        context associated with an authentication assertion.
      </xs:documentation>
    </xs:annotation>
  </xs:element>

  <xs:element name="Identification" type="IdentificationType">
    <xs:annotation>
      <xs:documentation>
        Refers to those characteristics that describe the
        processes and mechanisms
        the Authentication Authority uses to initially create
        an association between_u97 ? Principal
        and the identity (or name) by which the Principal will
        be known
      </xs:documentation>
    </xs:annotation>
  </xs:element>

  <xs:element name="PhysicalVerification">
    <xs:annotation>
      <xs:documentation>
        This element indicates_u116 that identification has been
        performed in a physical
        face-to-face meeting with the principal and not in an
        online manner.
      </xs:documentation>
    </xs:annotation>
    <xs:complexType>
      <xs:attribute name="credentialLevel">
        <xs:simpleType>
          <xs:restriction base="xs:NMTOKEN">
            <xs:enumeration value="primary"/>
            <xs:enumeration value="secondary"/>
          </xs:restriction>
        </xs:simpleType>
      </xs:attribute>
    </xs:complexType>
  </xs:element>

  <xs:element name="WrittenConsent" type="ExtensionOnlyType"/>

  <xs:element name="TechnicalProtection" type="TechnicalProtectionBaseType">
    <xs:annotation>
      <xs:documentation>
        Refers to those characterstics that describe how the
        'secret' (the knowledge or possession
        of which allows the Principal to authenticate to the
        Authentication Authority) is kept secure
      </xs:documentation>
    </xs:annotation>
  </xs:element>

  <xs:element name="SecretKeyProtection" type="SecretKeyProtectionType">
    <xs:annotation>
      <xs:documentation>
        This element indicates_u116 the types and strengths of
        facilities
        of a UA used to protect a shared secret key from
```

```
      unauthorized access and/or use.
    </xs:documentation>
  </xs:annotation>
</xs:element>

<xs:element name="PrivateKeyProtection" type="PrivateKeyProtectionType">
  <xs:annotation>
    <xs:documentation>
      This element indicates_u116 the types and strengths of
      facilities
      of a UA used to protect a private key from
      unauthorized access and/or use.
    </xs:documentation>
  </xs:annotation>
</xs:element>

<xs:element name="KeyActivation" type="KeyActivationType">
  <xs:annotation>
    <xs:documentation>The actions that must be performed
      before the private key_u99 can be used. </xs:documentation>
  </xs:annotation>
</xs:element>

<xs:element name="KeySharing" type="KeySharingType">
  <xs:annotation>
    <xs:documentation>Whether or not the private key_u105 is shared
      with the certificate authority.</xs:documentation>
  </xs:annotation>
</xs:element>

<xs:element name="KeyStorage" type="KeyStorageType">
  <xs:annotation>
    <xs:documentation>
      In which medium is the_u107 key stored.
      memory - the key is stored in memory.
      smartcard - the key is_u115 stored in a smartcard.
      token - the key is stored in a hardware token.
      MobileDevice - the key_u105 is stored in a mobile device.
      MobileAuthCard - the key is stored in a mobile
      authentication card.
    </xs:documentation>
  </xs:annotation>
</xs:element>

<xs:element name="SubscriberLineNumber" type="ExtensionOnlyType"/>
<xs:element name="UserSuffix" type="ExtensionOnlyType"/>

<xs:element name="Password" type="PasswordType">
  <xs:annotation>
    <xs:documentation>
      This element indicates_u116 that a password (or passphrase)
      has been used to
      authenticate the Principal to a remote system.
    </xs:documentation>
  </xs:annotation>
</xs:element>

<xs:element name="ActivationPin" type="ActivationPinType">
  <xs:annotation>
    <xs:documentation>
      This element indicates_u116 that a Pin (Personal
      Identification Number)_u104 has been used to authenticate the Principal
      to some local system in order to activate a key.
    </xs:documentation>
  </xs:annotation>
</xs:element>

<xs:element name="Token" type="TokenType">
  <xs:annotation>
    <xs:documentation>
      This element indicates_u116 that a hardware or software
```

```
      token is used
      as a method of identifying the Principal.
    </xs:documentation>
  </xs:annotation>
</xs:element>

<xs:element name="TimeSyncToken" type="TimeSyncTokenType">
  <xs:annotation>
    <xs:documentation>
      This element indicates_u116 that a time synchronization
      token is used to identify the Principal. hardware -
      the time synchonization
      token has been implemented in hardware. software - the
      time synchronization
      token has been implemented in software. SeedLength -
      the length, in bits, of the
      random seed used in the time synchronization token.
    </xs:documentation>
  </xs:annotation>
</xs:element>

<xs:element name="Smartcard" type="ExtensionOnlyType">
  <xs:annotation>
    <xs:documentation>
      This element indicates_u116 that a smartcard is used to
      identify the Principal.
    </xs:documentation>
  </xs:annotation>
</xs:element>

<xs:element name="Length" type="LengthType">
  <xs:annotation>
    <xs:documentation>
      This element indicates_u116 the minimum and/or maximum
      ASCII length of the password which is enforced (by the UA or the
      IdP). In other words, this is the minimum and/or maximum number of
      ASCII characters required to represent a valid password.
      min - the minimum number of ASCII characters required
      in a valid password, as enforced by the UA or the IdP.
      max - the maximum number of ASCII characters required
      in a valid password, as enforced by the UA or the IdP.
    </xs:documentation>
  </xs:annotation>
</xs:element>

<xs:element name="ActivationLimit" type="ActivationLimitType">
  <xs:annotation>
    <xs:documentation>
      This element indicates_u116 the length of time for which an
      PIN-based authentication is valid.
    </xs:documentation>
  </xs:annotation>
</xs:element>

<xs:element name="Generation">
  <xs:annotation>
    <xs:documentation>
      Indicates whether the password was chosen by the
      Principal or auto-supplied by the Authentication Authority.
      principal chosen - the Principal is allowed to choose
      the value of the password. This is true even if
      the initial password is chosen at random by the UA or
      the IdP and the Principal is then free to change
      the password.
      automatic - the password is chosen by the UA or the
      IdP to be cryptographically strong in some sense,
      or to satisfy certain password rules, and that the
      Principal is not free to change it or to choose a new password.
    </xs:documentation>
  </xs:annotation>
```

```
    <xs:complexType>
      <xs:attribute name="mechanism" use="required">
        <xs:simpleType>
          <xs:restriction base="xs:NMTOKEN">
            <xs:enumeration value="principalchosen"/>
            <xs:enumeration value="automatic"/>
          </xs:restriction>
        </xs:simpleType>
      </xs:attribute>
    </xs:complexType>
  </xs:element>

  <xs:element name="AuthnMethod" type="AuthnMethodBaseType">
    <xs:annotation>
      <xs:documentation>
        Refers to those characteristics that define the
        mechanisms by which the Principal authenticates to the Authentication
        Authority.
      </xs:documentation>
    </xs:annotation>
  </xs:element>

  <xs:element name="PrincipalAuthenticationMechanism"
type="PrincipalAuthenticationMechanismType">
    <xs:annotation>
      <xs:documentation>
        The method that a Principal employs to perform
        authentication to local system components.
      </xs:documentation>
    </xs:annotation>
  </xs:element>

  <xs:element name="Authenticator" type="AuthenticatorBaseType">
    <xs:annotation>
      <xs:documentation>
        The method applied to validate a principal's
        authentication across a network
      </xs:documentation>
    </xs:annotation>
  </xs:element>

  <xs:element name="ComplexAuthenticator" type="ComplexAuthenticatorType">
    <xs:annotation>
      <xs:documentation>
        Supports Authenticators with nested combinations of
        additional complexity.
      </xs:documentation>
    </xs:annotation>
  </xs:element>

  <xs:element name="PreviousSession" type="ExtensionOnlyType">
    <xs:annotation>
      <xs:documentation>
        Indicates that the Principal has been strongly
        authenticated in a previous session during which the IdP has set a
        cookie in the UA. During the present session the Principal has only
        been authenticated by the UA returning the cookie to the IdP.
      </xs:documentation>
    </xs:annotation>
  </xs:element>

  <xs:element name="ResumeSession" type="ExtensionOnlyType">
    <xs:annotation>
      <xs:documentation>
        Rather like PreviousSession but using stronger
        security. A secret that was established in a previous session with
        the Authentication Authority has been cached by the local system and
        is now re-used (e.g. a_u77 master Secret is used to derive new session
        keys in TLS, SSL, WTLS).
      </xs:documentation>
    </xs:annotation>
```

```
  </xs:element>

  <xs:element name="ZeroKnowledge" type="ExtensionOnlyType">
    <xs:annotation>
      <xs:documentation>
        This element indicates_u116 that the Principal has been
        authenticated by a zero knowledge technique as specified in ISO/IEC
        9798-5.
      </xs:documentation>
    </xs:annotation>
  </xs:element>

  <xs:element name="SharedSecretChallengeResponse"
type="SharedSecretChallengeResponseType"/>

  <xs:complexType name="SharedSecretChallengeResponseType">
    <xs:annotation>
      <xs:documentation>
        This element indicates_u116 that the Principal has been
        authenticated by a challenge-response protocol utilizing shared secret
        keys and symmetric cryptography.
      </xs:documentation>
    </xs:annotation>
    <xs:sequence>
      <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:attribute name="method" type="xs:anyURI" use="optional"/>
  </xs:complexType>

  <xs:element name="DigSig" type="PublicKeyType">
    <xs:annotation>
      <xs:documentation>
        This element indicates_u116 that the Principal has been
        authenticated by a mechanism which involves the Principal computing a
        digital signature over_u97 at least challenge data provided by the IdP.
      </xs:documentation>
    </xs:annotation>
  </xs:element>

  <xs:element name="AsymmetricDecryption" type="PublicKeyType">
    <xs:annotation>
      <xs:documentation>
        The local system has a_u112 Private key but it is used
        in decryption mode, rather than signature mode. For example, the
        Authentication Authority generates a secret and encrypts it using the
        local system's public key: the local system then proves it has
        decrypted the secret.
      </xs:documentation>
    </xs:annotation>
  </xs:element>

  <xs:element name="AsymmetricKeyAgreement" type="PublicKeyType">
    <xs:annotation>
      <xs:documentation>
        The local system has a_u112 Private key and uses it for
        shared secret key agreement with the Authentication Authority (e.g.
        via Diffie Helman).
      </xs:documentation>
    </xs:annotation>
  </xs:element>

  <xs:complexType name="PublicKeyType">
    <xs:sequence>
      <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:attribute name="keyValidation" use="optional"/>
  </xs:complexType>

  <xs:element name="IPAddress" type="ExtensionOnlyType">
    <xs:annotation>
      <xs:documentation>
```

```
        This element indicates_u116 that the Principal has been
        authenticated through connection from a particular IP address.
      </xs:documentation>
    </xs:annotation>
  </xs:element>

  <xs:element name="SharedSecretDynamicPlaintext" type="ExtensionOnlyType">
    <xs:annotation>
      <xs:documentation>
        The local system and Authentication Authority
        share a secret key. The local system uses this to encrypt a
        randomised string to pass to the Authentication Authority.
      </xs:documentation>
    </xs:annotation>
  </xs:element>

  <xs:element name="AuthenticatorTransportProtocol"
type="AuthenticatorTransportProtocolType">
    <xs:annotation>
      <xs:documentation>
        The protocol across which Authenticator information is
        transferred to an Authentication Authority verifier.
      </xs:documentation>
    </xs:annotation>
  </xs:element>

  <xs:element name="HTTP" type="ExtensionOnlyType">
    <xs:annotation>
      <xs:documentation>
        This element indicates_u116 that the Authenticator has been
        transmitted using bare_u72 HTTP utilizing no additional security
        protocols.
      </xs:documentation>
    </xs:annotation>
  </xs:element>

  <xs:element name="IPSec" type="ExtensionOnlyType">
    <xs:annotation>
      <xs:documentation>
        This element indicates_u116 that the Authenticator has been
        transmitted using a transport mechanism protected by an IPSEC session.
      </xs:documentation>
    </xs:annotation>
  </xs:element>

  <xs:element name="WTLS" type="ExtensionOnlyType">
    <xs:annotation>
      <xs:documentation>
        This element indicates_u116 that the Authenticator has been
        transmitted using a transport mechanism protected by a WTLS session.
      </xs:documentation>
    </xs:annotation>
  </xs:element>

  <xs:element name="MobileNetworkNoEncryption" type="ExtensionOnlyType">
    <xs:annotation>
      <xs:documentation>
        This element indicates_u116 that the Authenticator has been
        transmitted solely across a mobile network using no additional
        security mechanism.
      </xs:documentation>
    </xs:annotation>
  </xs:element>

  <xs:element name="MobileNetworkRadioEncryption" type="ExtensionOnlyType"/>
  <xs:element name="MobileNetworkEndToEndEncryption" type="ExtensionOnlyType"/>

  <xs:element name="SSL" type="ExtensionOnlyType">
    <xs:annotation>
      <xs:documentation>
        This element indicates_u116 that the Authenticator has been
```

```
      transmitted using a transport mechanism protected by an SSL or TLS
      session.
    </xs:documentation>
  </xs:annotation>
</xs:element>

<xs:element name="PSTN" type="ExtensionOnlyType"/>
<xs:element name="ISDN" type="ExtensionOnlyType"/>
<xs:element name="ADSL" type="ExtensionOnlyType"/>

<xs:element name="OperationalProtection" type="OperationalProtectionType">
  <xs:annotation>
    <xs:documentation>
      Refers to those characteristics that describe
      procedural security controls employed by the Authentication Authority.
    </xs:documentation>
  </xs:annotation>
</xs:element>

<xs:element name="SecurityAudit" type="SecurityAuditType"/>
<xs:element name="SwitchAudit" type="ExtensionOnlyType"/>
<xs:element name="DeactivationCallCenter" type="ExtensionOnlyType"/>

<xs:element name="GoverningAgreements" type="GoverningAgreementsType">
  <xs:annotation>
    <xs:documentation>
      Provides a mechanism for linking to external (likely
      human readable) documents in which additional business agreements,
      (e.g. liability constraints, obligations, etc.) can be placed.
    </xs:documentation>
  </xs:annotation>
</xs:element>

<xs:element name="GoverningAgreementRef" type="GoverningAgreementRefType"/>

<xs:simpleType name="nymType">
  <xs:restriction base="xs:NMTOKEN">
    <xs:enumeration value="anonymity"/>
    <xs:enumeration value="verinymity"/>
    <xs:enumeration value="pseudonymity"/>
  </xs:restriction>
</xs:simpleType>

<xs:complexType name="AuthnContextDeclarationBaseType">
  <xs:sequence>
    <xs:element ref="Identification" minOccurs="0"/>
    <xs:element ref="TechnicalProtection" minOccurs="0"/>
    <xs:element ref="OperationalProtection" minOccurs="0"/>
    <xs:element ref="AuthnMethod" minOccurs="0"/>
    <xs:element ref="GoverningAgreements" minOccurs="0"/>
    <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
  </xs:sequence>
  <xs:attribute name="ID" type="xs:ID" use="optional"/>
</xs:complexType>

<xs:complexType name="IdentificationType">
  <xs:sequence>
    <xs:element ref="PhysicalVerification" minOccurs="0"/>
    <xs:element ref="WrittenConsent" minOccurs="0"/>
    <xs:element ref="GoverningAgreements" minOccurs="0"/>
    <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
  </xs:sequence>
  <xs:attribute name="nym" type="nymType">
    <xs:annotation>
      <xs:documentation>
        This attribute indicates whether or not the
        Identification mechanisms allow the actions of the Principal to be
        linked to an actual end user.
      </xs:documentation>
    </xs:annotation>
  </xs:attribute>
```

```
</xs:complexType>

<xs:complexType name="TechnicalProtectionBaseType">
  <xs:sequence>
    <xs:choice minOccurs="0">
      <xs:element ref="PrivateKeyProtection"/>
      <xs:element ref="SecretKeyProtection"/>
    </xs:choice>
    <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>

<xs:complexType name="OperationalProtectionType">
  <xs:sequence>
    <xs:element ref="SecurityAudit" minOccurs="0"/>
    <xs:element ref="DeactivationCallCenter" minOccurs="0"/>
    <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>

<xs:complexType name="AuthnMethodBaseType">
  <xs:sequence>
    <xs:element ref="PrincipalAuthenticationMechanism" minOccurs="0"/>
    <xs:element ref="Authenticator" minOccurs="0"/>
    <xs:element ref="AuthenticatorTransportProtocol" minOccurs="0"/>
    <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>

<xs:complexType name="GoverningAgreementsType">
  <xs:sequence>
    <xs:element ref="GoverningAgreementRef" maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>

<xs:complexType name="GoverningAgreementRefType">
  <xs:attribute name="governingAgreementRef" type="xs:anyURI" use="required"/>
</xs:complexType>

<xs:complexType name="PrincipalAuthenticationMechanismType">
  <xs:sequence>
    <xs:element ref="Password" minOccurs="0"/>
    <xs:element ref="RestrictedPassword" minOccurs="0"/>
    <xs:element ref="Token" minOccurs="0"/>
    <xs:element ref="Smartcard" minOccurs="0"/>
    <xs:element ref="ActivationPin" minOccurs="0"/>
    <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
  </xs:sequence>
  <xs:attribute name="preauth" type="xs:integer" use="optional"/>
</xs:complexType>

<xs:group name="AuthenticatorChoiceGroup">
  <xs:choice>
    <xs:element ref="PreviousSession"/>
    <xs:element ref="ResumeSession"/>
    <xs:element ref="DigSig"/>
    <xs:element ref="Password"/>
    <xs:element ref="RestrictedPassword"/>
    <xs:element ref="ZeroKnowledge"/>
    <xs:element ref="SharedSecretChallengeResponse"/>
    <xs:element ref="SharedSecretDynamicPlaintext"/>
    <xs:element ref="IPAddress"/>
    <xs:element ref="AsymmetricDecryption"/>
    <xs:element ref="AsymmetricKeyAgreement"/>
    <xs:element ref="SubscriberLineNumber"/>
    <xs:element ref="UserSuffix"/>
    <xs:element ref="ComplexAuthenticator"/>
  </xs:choice>
</xs:group>

<xs:group name="AuthenticatorSequenceGroup">
```

```
    <xs:sequence>
      <xs:element ref="PreviousSession" minOccurs="0"/>
      <xs:element ref="ResumeSession" minOccurs="0"/>
      <xs:element ref="DigSig" minOccurs="0"/>
      <xs:element ref="Password" minOccurs="0"/>
      <xs:element ref="RestrictedPassword" minOccurs="0"/>
      <xs:element ref="ZeroKnowledge" minOccurs="0"/>
      <xs:element ref="SharedSecretChallengeResponse" minOccurs="0"/>
      <xs:element ref="SharedSecretDynamicPlaintext" minOccurs="0"/>
      <xs:element ref="IPAddress" minOccurs="0"/>
      <xs:element ref="AsymmetricDecryption" minOccurs="0"/>
      <xs:element ref="AsymmetricKeyAgreement" minOccurs="0"/>
      <xs:element ref="SubscriberLineNumber" minOccurs="0"/>
      <xs:element ref="UserSuffix" minOccurs="0"/>
      <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:group>

  <xs:complexType name="AuthenticatorBaseType">
    <xs:sequence>
      <xs:group ref="AuthenticatorChoiceGroup"/>
      <xs:group ref="AuthenticatorSequenceGroup"/>
    </xs:sequence>
  </xs:complexType>

  <xs:complexType name="ComplexAuthenticatorType">
    <xs:sequence>
      <xs:group ref="AuthenticatorChoiceGroup"/>
      <xs:group ref="AuthenticatorSequenceGroup"/>
    </xs:sequence>
  </xs:complexType>

  <xs:complexType name="AuthenticatorTransportProtocolType">
    <xs:sequence>
      <xs:choice minOccurs="0">
        <xs:element ref="HTTP"/>
        <xs:element ref="SSL"/>
        <xs:element ref="MobileNetworkNoEncryption"/>
        <xs:element ref="MobileNetworkRadioEncryption"/>
        <xs:element ref="MobileNetworkEndToEndEncryption"/>
        <xs:element ref="WTLS"/>
        <xs:element ref="IPSec"/>
        <xs:element ref="PSTN"/>
        <xs:element ref="ISDN"/>
        <xs:element ref="ADSL"/>
      </xs:choice>
      <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>

  <xs:complexType name="KeyActivationType">
    <xs:sequence>
      <xs:element ref="ActivationPin" minOccurs="0"/>
      <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>

  <xs:complexType name="KeySharingType">
    <xs:attribute name="sharing" type="xs:boolean" use="required"/>
  </xs:complexType>

  <xs:complexType name="PrivateKeyProtectionType">
    <xs:sequence>
      <xs:element ref="KeyActivation" minOccurs="0"/>
      <xs:element ref="KeyStorage" minOccurs="0"/>
      <xs:element ref="KeySharing" minOccurs="0"/>
      <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>

  <xs:complexType name="PasswordType">
```

```
    <xs:sequence>
      <xs:element ref="Length" minOccurs="0"/>
      <xs:element ref="Alphabet" minOccurs="0"/>
      <xs:element ref="Generation" minOccurs="0"/>
      <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:attribute name="ExternalVerification" type="xs:anyURI" use="optional"/>
  </xs:complexType>

  <xs:element name="RestrictedPassword" type="RestrictedPasswordType"/>

  <xs:complexType name="RestrictedPasswordType">
    <xs:complexContent>
      <xs:restriction base="PasswordType">
        <xs:sequence>
          <xs:element name="Length" type="RestrictedLengthType" minOccurs="1"/>
          <xs:element ref="Generation" minOccurs="0"/>
          <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
        </xs:sequence>
        <xs:attribute name="ExternalVerification" type="xs:anyURI"
use="optional"/>
      </xs:restriction>
    </xs:complexContent>
  </xs:complexType>

  <xs:complexType name="RestrictedLengthType">
    <xs:complexContent>
      <xs:restriction base="LengthType">
        <xs:attribute name="min" use="required">
          <xs:simpleType>
            <xs:restriction base="xs:integer">
              <xs:minInclusive value="3"/>
            </xs:restriction>
          </xs:simpleType>
        </xs:attribute>
        <xs:attribute name="max" type="xs:integer" use="optional"/>
      </xs:restriction>
    </xs:complexContent>
  </xs:complexType>

  <xs:complexType name="ActivationPinType">
    <xs:sequence>
      <xs:element ref="Length" minOccurs="0"/>
      <xs:element ref="Alphabet" minOccurs="0"/>
      <xs:element ref="Generation" minOccurs="0"/>
      <xs:element ref="ActivationLimit" minOccurs="0"/>
      <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>

  <xs:element name="Alphabet" type="AlphabetType"/>
  <xs:complexType name="AlphabetType">
    <xs:attribute name="requiredChars" type="xs:string" use="required"/>
    <xs:attribute name="excludedChars" type="xs:string" use="optional"/>
    <xs:attribute name="case" type="xs:string" use="optional"/>
  </xs:complexType>

  <xs:complexType name="TokenType">
    <xs:sequence>
      <xs:element ref="TimeSyncToken"/>
      <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>

  <xs:simpleType name="DeviceTypeType">
    <xs:restriction base="xs:NMTOKEN">
      <xs:enumeration value="hardware"/>
      <xs:enumeration value="software"/>
    </xs:restriction>
  </xs:simpleType>
```

```
<xs:simpleType name="booleanType">
  <xs:restriction base="xs:NMTOKEN">
    <xs:enumeration value="true"/>
    <xs:enumeration value="false"/>
  </xs:restriction>
</xs:simpleType>

<xs:complexType name="TimeSyncTokenType">
  <xs:attribute name="DeviceType" type="DeviceTypeType" use="required"/>
  <xs:attribute name="SeedLength" type="xs:integer" use="required"/>
  <xs:attribute name="DeviceInHand" type="booleanType" use="required"/>
</xs:complexType>

<xs:complexType name="ActivationLimitType">
  <xs:choice>
    <xs:element ref="ActivationLimitDuration"/>
    <xs:element ref="ActivationLimitUsages"/>
    <xs:element ref="ActivationLimitSession"/>
  </xs:choice>
</xs:complexType>

<xs:element name="ActivationLimitDuration" type="ActivationLimitDurationType">
  <xs:annotation>
    <xs:documentation>
      This element indicates_u116 that the Key Activation Limit is
      defined as a specific duration of time.
    </xs:documentation>
  </xs:annotation>
</xs:element>

<xs:element name="ActivationLimitUsages" type="ActivationLimitUsagesType">
  <xs:annotation>
    <xs:documentation>
      This element indicates_u116 that the Key Activation Limit is
      defined as a number of_u117 usages.
    </xs:documentation>
  </xs:annotation>
</xs:element>

<xs:element name="ActivationLimitSession" type="ActivationLimitSessionType">
  <xs:annotation>
    <xs:documentation>
      This element indicates_u116 that the Key Activation Limit is
      the session.
    </xs:documentation>
  </xs:annotation>
</xs:element>

<xs:complexType name="ActivationLimitDurationType">
  <xs:attribute name="duration" type="xs:duration" use="required"/>
</xs:complexType>

<xs:complexType name="ActivationLimitUsagesType">
  <xs:attribute name="number" type="xs:integer" use="required"/>
</xs:complexType>

<xs:complexType name="ActivationLimitSessionType"/>

<xs:complexType name="LengthType">
  <xs:attribute name="min" type="xs:integer" use="required"/>
  <xs:attribute name="max" type="xs:integer" use="optional"/>
</xs:complexType>

<xs:simpleType name="mediumType">
  <xs:restriction base="xs:NMTOKEN">
    <xs:enumeration value="memory"/>
    <xs:enumeration value="smartcard"/>
    <xs:enumeration value="token"/>
    <xs:enumeration value="MobileDevice"/>
    <xs:enumeration value="MobileAuthCard"/>
  </xs:restriction>
```

```
  </xs:simpleType>

  <xs:complexType name="KeyStorageType">
    <xs:attribute name="medium" type="mediumType" use="required"/>
  </xs:complexType>

  <xs:complexType name="SecretKeyProtectionType">
    <xs:sequence>
      <xs:element ref="KeyActivation" minOccurs="0"/>
      <xs:element ref="KeyStorage" minOccurs="0"/>
      <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>

  <xs:complexType name="SecurityAuditType">
    <xs:sequence>
      <xs:element ref="SwitchAudit" minOccurs="0"/>
      <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>

  <xs:complexType name="ExtensionOnlyType">
    <xs:sequence>
      <xs:element ref="Extension" minOccurs="0"  maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>

  <xs:element name="Extension" type="ExtensionType"/>

  <xs:complexType name="ExtensionType">
    <xs:sequence>
      <xs:any namespace="##other" processContents="lax" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>

</xs:schema>
```

A.25 Schéma de contexte d'authentification SAML X.509

C'est le schéma de contexte d'authentification SAML X.509.

```
<?xml version="1.0" encoding="UTF-8"?>

<xs:schema targetNamespace="urn:oasis:names:tc:SAML:2.0:ac:classes:X509"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns="urn:oasis:names:tc:SAML:2.0:ac:classes:X509"
  finalDefault="extension"
  blockDefault="substitution"
  version="2.0">

  <xs:redefine schemaLocation="saml-schema-authn-context-types-2.0.xsd">

    <xs:annotation>
      <xs:documentation>
        Class identifier: urn:oasis:names:tc:SAML:2.0:ac:classes:X509
        Document identifier: saml-schema-authn-context-x509-2.0
        Location: http://docs.oasis-open.org/security/saml/v2.0/
        Revision history:
          V2.0 (March, 2005):
            New authentication_u99 context class schema for SAML V2.0.
      </xs:documentation>
    </xs:annotation>

    <xs:complexType name="AuthnContextDeclarationBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthnContextDeclarationBaseType">
          <xs:sequence>
            <xs:element ref="Identification" minOccurs="0"/>
            <xs:element ref="TechnicalProtection" minOccurs="0"/>
```

```
            <xs:element ref="OperationalProtection" minOccurs="0"/>
            <xs:element ref="AuthnMethod"/>
            <xs:element ref="GoverningAgreements" minOccurs="0"/>
            <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
          </xs:sequence>
          <xs:attribute name="ID" type="xs:ID" use="optional"/>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="AuthnMethodBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthnMethodBaseType">
          <xs:sequence>
            <xs:element ref="PrincipalAuthenticationMechanism"/>
            <xs:element ref="Authenticator"/>
            <xs:element ref="AuthenticatorTransportProtocol" minOccurs="0"/>
            <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="PrincipalAuthenticationMechanismType">
      <xs:complexContent>
        <xs:restriction base="PrincipalAuthenticationMechanismType">
          <xs:sequence>
            <xs:element ref="RestrictedPassword"/>
          </xs:sequence>
          <xs:attribute name="preauth" type="xs:integer" use="optional"/>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="AuthenticatorBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthenticatorBaseType">
          <xs:sequence>
            <xs:element ref="DigSig"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="PublicKeyType">
      <xs:complexContent>
        <xs:restriction base="PublicKeyType">
          <xs:attribute name="keyValidation" type="xs:anyURI"
fixed="urn:oasis:names:tc:SAML:2.0:ac:classes:X509"/>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

  </xs:redefine>

</xs:schema>
```

A.26 Schéma SAML du contexte d'authentification XMLDSig

C'est le schéma de contexte d'authentification SAML de signature numérique XML.

```
<?xml version="1.0" encoding="UTF-8"?>

<xs:schema targetNamespace="urn:oasis:names:tc:SAML:2.0:ac:classes:XMLDSig"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns="urn:oasis:names:tc:SAML:2.0:ac:classes:XMLDSig"
  finalDefault="extension"
  blockDefault="substitution"
  version="2.0">
```

```
  <xs:redefine schemaLocation="saml-schema-authn-context-types-2.0.xsd">

    <xs:annotation>
      <xs:documentation>
        Class identifier: urn:oasis:names:tc:SAML:2.0:ac:classes:XMLDSig
        Document identifier: saml-schema-authn-context-xmldsig-2.0
        Location: http://docs.oasis-open.org/security/saml/v2.0/
        Revision history:
          V2.0 (March, 2005):
            New authentication_u99 context class schema for SAML V2.0.
      </xs:documentation>
    </xs:annotation>

    <xs:complexType name="AuthnContextDeclarationBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthnContextDeclarationBaseType">
          <xs:sequence>
            <xs:element ref="Identification" minOccurs="0"/>
            <xs:element ref="TechnicalProtection" minOccurs="0"/>
            <xs:element ref="OperationalProtection" minOccurs="0"/>
            <xs:element ref="AuthnMethod"/>
            <xs:element ref="GoverningAgreements" minOccurs="0"/>
            <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
          </xs:sequence>
          <xs:attribute name="ID" type="xs:ID" use="optional"/>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="AuthnMethodBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthnMethodBaseType">
          <xs:sequence>
            <xs:element ref="PrincipalAuthenticationMechanism"/>
            <xs:element ref="Authenticator"/>
            <xs:element ref="AuthenticatorTransportProtocol" minOccurs="0"/>
            <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="PrincipalAuthenticationMechanismType">
      <xs:complexContent>
        <xs:restriction base="PrincipalAuthenticationMechanismType">
          <xs:sequence>
            <xs:element ref="RestrictedPassword"/>
          </xs:sequence>
          <xs:attribute name="preauth" type="xs:integer" use="optional"/>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="AuthenticatorBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthenticatorBaseType">
          <xs:sequence>
            <xs:element ref="DigSig"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="PublicKeyType">
      <xs:complexContent>
        <xs:restriction base="PublicKeyType">
          <xs:attribute name="keyValidation" type="xs:anyURI"
fixed="urn:ietf:rfc:3075"/>
        </xs:restriction>
      </xs:complexContent>
```

```
    </xs:complexType>

  </xs:redefine>

</xs:schema>
```

A.27 Schéma SAML d'ECP

C'est le schéma SAML qui fait la liste des profils de client ou mandataire amélioré (ECP, *enhanced client or proxy*).

```
<?xml version="1.0" encoding="UTF-8"?>
<schema
    targetNamespace="urn:oasis:names:tc:SAML:2.0:profiles:SSO:ecp"
    xmlns="http://www.w3.org/2001/XMLSchema"
    xmlns:ecp="urn:oasis:names:tc:SAML:2.0:profiles:SSO:ecp"
    xmlns:samlp="urn:oasis:names:tc:SAML:2.0:protocol"
    xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion"
    xmlns:S="http://schemas.xmlsoap.org/soap/envelope/"
    elementFormDefault="unqualified"
    attributeFormDefault="unqualified"
    blockDefault="substitution"
    version="2.0">
    <import namespace="urn:oasis:names:tc:SAML:2.0:protocol"
        schemaLocation="saml-schema-protocol-2.0.xsd"/>
    <import namespace="urn:oasis:names:tc:SAML:2.0:assertion"
        schemaLocation="saml-schema-assertion-2.0.xsd"/>
    <import namespace="http://schemas.xmlsoap.org/soap/envelope/"
        schemaLocation="http://schemas.xmlsoap.org/soap/envelope/"/>
    <annotation>
        <documentation>
            Document identifier: saml-schema-ecp-2.0
            Location: http://docs.oasis-open.org/security/saml/v2.0/
            Revision history:
              V2.0 (March, 2005):
                Custom schema for ECP profile, first published in SAML 2.0.
        </documentation>
    </annotation>

    <element name="Request" type="ecp:RequestType"/>
    <complexType name="RequestType">
        <sequence>
            <element ref="saml:Issuer"/>
            <element ref="samlp:IDPList" minOccurs="0"/>
        </sequence>
        <attribute ref="S:mustUnderstand" use="required"/>
        <attribute ref="S:actor" use="required"/>
        <attribute name="ProviderName" type="string" use="optional"/>
        <attribute name="IsPassive" type="boolean" use="optional"/>
    </complexType>

    <element name="Response" type="ecp:ResponseType"/>
    <complexType name="ResponseType">
        <attribute ref="S:mustUnderstand" use="required"/>
        <attribute ref="S:actor" use="required"/>
        <attribute name="AssertionConsumerServiceURL" type="anyURI"
use="required"/>
    </complexType>

    <element name="RelayState" type="ecp:RelayStateType"/>
    <complexType name="RelayStateType">
        <simpleContent>
            <extension base="string">
                <attribute ref="S:mustUnderstand" use="required"/>
                <attribute ref="S:actor" use="required"/>
            </extension>
        </simpleContent>
    </complexType>
</schema>
```

A.28 Schéma SAML de métadonnées

C'est la liste pour le schéma de métadonnées SAML.

```
<?xml version="1.0" encoding="UTF-8"?>
<schema
    targetNamespace="urn:oasis:names:tc:SAML:2.0:metadata"
    xmlns:md="urn:oasis:names:tc:SAML:2.0:metadata"
    xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
    xmlns:xenc="http://www.w3.org/2001/04/xmlenc#"
    xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion"
    xmlns="http://www.w3.org/2001/XMLSchema"
    elementFormDefault="unqualified"
    attributeFormDefault="unqualified"
    blockDefault="substitution"
    version="2.0">
    <import namespace="http://www.w3.org/2000/09/xmldsig#"
        schemaLocation="http://www.w3.org/TR/2002/REC-xmldsig-core-
20020212/xmldsig-core-schema.xsd"/>
    <import namespace="http://www.w3.org/2001/04/xmlenc#"
        schemaLocation="http://www.w3.org/TR/2002/REC-xmlenc-core-20021210/xenc-
schema.xsd"/>
    <import namespace="urn:oasis:names:tc:SAML:2.0:assertion"
        schemaLocation="saml-schema-assertion-2.0.xsd"/>
    <import namespace="http://www.w3.org/XML/1998/namespace"
        schemaLocation="http://www.w3.org/2001/xml.xsd"/>
    <annotation>
        <documentation>
            Document identifier: saml-schema-metadata-2.0
            Location: http://docs.oasis-open.org/security/saml/v2.0/
            Revision history:
              V2.0 (March, 2005):
                Schema for SAML metadata, first published in SAML 2.0.
        </documentation>
    </annotation>

    <simpleType name="entityIDType">
        <restriction base="anyURI">
            <maxLength value="1024"/>
        </restriction>
    </simpleType>
    <complexType name="localizedNameType">
        <simpleContent>
            <extension base="string">
                <attribute ref="xml:lang" use="required"/>
            </extension>
        </simpleContent>
    </complexType>
    <complexType name="localizedURIType">
        <simpleContent>
            <extension base="anyURI">
                <attribute ref="xml:lang" use="required"/>
            </extension>
        </simpleContent>
    </complexType>

    <element name="Extensions" type="md:ExtensionsType"/>
    <complexType final="#all" name="ExtensionsType">
        <sequence>
            <any namespace="##other" processContents="lax"
maxOccurs="unbounded"/>
        </sequence>
    </complexType>

    <complexType name="EndpointType">
        <sequence>
            <any namespace="##other" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
        </sequence>
        <attribute name="Binding" type="anyURI" use="required"/>
```

```
        <attribute name="Location" type="anyURI" use="required"/>
        <attribute name="ResponseLocation" type="anyURI" use="optional"/>
        <anyAttribute namespace="##other" processContents="lax"/>
    </complexType>

    <complexType name="IndexedEndpointType">
        <complexContent>
            <extension base="md:EndpointType">
                <attribute name="index" type="unsignedShort" use="required"/>
                <attribute name="isDefault" type="boolean" use="optional"/>
            </extension>
        </complexContent>
    </complexType>

    <element name="EntitiesDescriptor" type="md:EntitiesDescriptorType"/>
    <complexType name="EntitiesDescriptorType">
        <sequence>
            <element ref="ds:Signature" minOccurs="0"/>
            <element ref="md:Extensions" minOccurs="0"/>
            <choice minOccurs="1" maxOccurs="unbounded">
                <element ref="md:EntityDescriptor"/>
                <element ref="md:EntitiesDescriptor"/>
            </choice>
        </sequence>
        <attribute name="validUntil" type="dateTime" use="optional"/>
        <attribute name="cacheDuration" type="duration" use="optional"/>
        <attribute name="ID" type="ID" use="optional"/>
        <attribute name="Name" type="string" use="optional"/>
    </complexType>

    <element name="EntityDescriptor" type="md:EntityDescriptorType"/>
    <complexType name="EntityDescriptorType">
        <sequence>
            <element ref="ds:Signature" minOccurs="0"/>
            <element ref="md:Extensions" minOccurs="0"/>
            <choice>
                <choice maxOccurs="unbounded">
                    <element ref="md:RoleDescriptor"/>
                    <element ref="md:IDPSSODescriptor"/>
                    <element ref="md:SPSSODescriptor"/>
                    <element ref="md:AuthnAuthorityDescriptor"/>
                    <element ref="md:AttributeAuthorityDescriptor"/>
                    <element ref="md:PDPDescriptor"/>
                </choice>
                <element ref="md:AffiliationDescriptor"/>
            </choice>
            <element ref="md:Organization" minOccurs="0"/>
            <element ref="md:ContactPerson" minOccurs="0" maxOccurs="unbounded"/>
            <element ref="md:AdditionalMetadataLocation" minOccurs="0"
maxOccurs="unbounded"/>
        </sequence>
        <attribute name="entityID" type="md:entityIDType" use="required"/>
        <attribute name="validUntil" type="dateTime" use="optional"/>
        <attribute name="cacheDuration" type="duration" use="optional"/>
        <attribute name="ID" type="ID" use="optional"/>
        <anyAttribute namespace="##other" processContents="lax"/>
    </complexType>

    <element name="Organization" type="md:OrganizationType"/>
    <complexType name="OrganizationType">
        <sequence>
            <element ref="md:Extensions" minOccurs="0"/>
            <element ref="md:OrganizationName" maxOccurs="unbounded"/>
            <element ref="md:OrganizationDisplayName" maxOccurs="unbounded"/>
            <element ref="md:OrganizationURL" maxOccurs="unbounded"/>
        </sequence>
        <anyAttribute namespace="##other" processContents="lax"/>
    </complexType>
    <element name="OrganizationName" type="md:localizedNameType"/>
    <element name="OrganizationDisplayName" type="md:localizedNameType"/>
    <element name="OrganizationURL" type="md:localizedURIType"/>
```

```
    <element name="ContactPerson" type="md:ContactType"/>
    <complexType name="ContactType">
        <sequence>
            <element ref="md:Extensions" minOccurs="0"/>
            <element ref="md:Company" minOccurs="0"/>
            <element ref="md:GivenName" minOccurs="0"/>
            <element ref="md:SurName" minOccurs="0"/>
            <element ref="md:EmailAddress" minOccurs="0" maxOccurs="unbounded"/>
            <element ref="md:TelephoneNumber" minOccurs="0"
maxOccurs="unbounded"/>
        </sequence>
        <attribute name="contactType" type="md:ContactTypeType" use="required"/>
        <anyAttribute namespace="##other" processContents="lax"/>
    </complexType>
    <element name="Company" type="string"/>
    <element name="GivenName" type="string"/>
    <element name="SurName" type="string"/>
    <element name="EmailAddress" type="anyURI"/>
    <element name="TelephoneNumber" type="string"/>
    <simpleType name="ContactTypeType">
        <restriction base="string">
            <enumeration value="technical"/>
            <enumeration value="support"/>
            <enumeration value="administrative"/>
            <enumeration value="billing"/>
            <enumeration value="other"/>
        </restriction>
    </simpleType>

    <element name="AdditionalMetadataLocation"
type="md:AdditionalMetadataLocationType"/>
    <complexType name="AdditionalMetadataLocationType">
        <simpleContent>
            <extension base="anyURI">
                <attribute name="namespace" type="anyURI" use="required"/>
            </extension>
        </simpleContent>
    </complexType>

    <element name="RoleDescriptor" type="md:RoleDescriptorType"/>
    <complexType name="RoleDescriptorType" abstract="true">
        <sequence>
            <element ref="ds:Signature" minOccurs="0"/>
            <element ref="md:Extensions" minOccurs="0"/>
            <element ref="md:KeyDescriptor" minOccurs="0" maxOccurs="unbounded"/>
            <element ref="md:Organization" minOccurs="0"/>
            <element ref="md:ContactPerson" minOccurs="0" maxOccurs="unbounded"/>
        </sequence>
        <attribute name="ID" type="ID" use="optional"/>
        <attribute name="validUntil" type="dateTime" use="optional"/>
        <attribute name="cacheDuration" type="duration" use="optional"/>
        <attribute name="protocolSupportEnumeration" type="md:anyURIListType"
use="required"/>
        <attribute name="errorURL" type="anyURI" use="optional"/>
        <anyAttribute namespace="##other" processContents="lax"/>
    </complexType>
    <simpleType name="anyURIListType">
        <list itemType="anyURI"/>
    </simpleType>

    <element name="KeyDescriptor" type="md:KeyDescriptorType"/>
    <complexType name="KeyDescriptorType">
        <sequence>
            <element ref="ds:KeyInfo"/>
            <element ref="md:EncryptionMethod" minOccurs="0"
maxOccurs="unbounded"/>
        </sequence>
        <attribute name="use" type="md:KeyTypes" use="optional"/>
    </complexType>
    <simpleType name="KeyTypes">
        <restriction base="string">
```

```
            <enumeration value="encryption"/>
            <enumeration value="signing"/>
        </restriction>
    </simpleType>
    <element name="EncryptionMethod" type="xenc:EncryptionMethodType"/>

    <complexType name="SSODescriptorType" abstract="true">
        <complexContent>
            <extension base="md:RoleDescriptorType">
                <sequence>
                    <element ref="md:ArtifactResolutionService" minOccurs="0"
maxOccurs="unbounded"/>
                    <element ref="md:SingleLogoutService" minOccurs="0"
maxOccurs="unbounded"/>
                    <element ref="md:ManageNameIDService" minOccurs="0"
maxOccurs="unbounded"/>
                    <element ref="md:NameIDFormat" minOccurs="0"
maxOccurs="unbounded"/>
                </sequence>
            </extension>
        </complexContent>
    </complexType>
    <element name="ArtifactResolutionService" type="md:IndexedEndpointType"/>
    <element name="SingleLogoutService" type="md:EndpointType"/>
    <element name="ManageNameIDService" type="md:EndpointType"/>
    <element name="NameIDFormat" type="anyURI"/>

    <element name="IDPSSODescriptor" type="md:IDPSSODescriptorType"/>
    <complexType name="IDPSSODescriptorType">
        <complexContent>
            <extension base="md:SSODescriptorType">
                <sequence>
                    <element ref="md:SingleSignOnService" maxOccurs="unbounded"/>
                    <element ref="md:NameIDMappingService" minOccurs="0"
maxOccurs="unbounded"/>
                    <element ref="md:AssertionIDRequestService" minOccurs="0"
maxOccurs="unbounded"/>
                    <element ref="md:AttributeProfile" minOccurs="0"
maxOccurs="unbounded"/>
                    <element ref="saml:Attribute" minOccurs="0"
maxOccurs="unbounded"/>
                </sequence>
                <attribute name="WantAuthnRequestsSigned" type="boolean"
use="optional"/>
            </extension>
        </complexContent>
    </complexType>
    <element name="SingleSignOnService" type="md:EndpointType"/>
    <element name="NameIDMappingService" type="md:EndpointType"/>
    <element name="AssertionIDRequestService" type="md:EndpointType"/>
    <element name="AttributeProfile" type="anyURI"/>

    <element name="SPSSODescriptor" type="md:SPSSODescriptorType"/>
    <complexType name="SPSSODescriptorType">
        <complexContent>
            <extension base="md:SSODescriptorType">
                <sequence>
                    <element ref="md:AssertionConsumerService"
maxOccurs="unbounded"/>
                    <element ref="md:AttributeConsumingService" minOccurs="0"
maxOccurs="unbounded"/>
                </sequence>
                <attribute name="AuthnRequestsSigned" type="boolean"
use="optional"/>
                <attribute name="WantAssertionsSigned" type="boolean"
use="optional"/>
            </extension>
        </complexContent>
    </complexType>
    <element name="AssertionConsumerService" type="md:IndexedEndpointType"/>
```

```
    <element name="AttributeConsumingService"
type="md:AttributeConsumingServiceType"/>
    <complexType name="AttributeConsumingServiceType">
        <sequence>
            <element ref="md:ServiceName" maxOccurs="unbounded"/>
            <element ref="md:ServiceDescription" minOccurs="0"
maxOccurs="unbounded"/>
            <element ref="md:RequestedAttribute" maxOccurs="unbounded"/>
        </sequence>
        <attribute name="index" type="unsignedShort" use="required"/>
        <attribute name="isDefault" type="boolean" use="optional"/>
    </complexType>
    <element name="ServiceName" type="md:localizedNameType"/>
    <element name="ServiceDescription" type="md:localizedNameType"/>
    <element name="RequestedAttribute" type="md:RequestedAttributeType"/>
    <complexType name="RequestedAttributeType">
        <complexContent>
            <extension base="saml:AttributeType">
                <attribute name="isRequired" type="boolean" use="optional"/>
            </extension>
        </complexContent>
    </complexType>

    <element name="AuthnAuthorityDescriptor"
type="md:AuthnAuthorityDescriptorType"/>
    <complexType name="AuthnAuthorityDescriptorType">
        <complexContent>
            <extension base="md:RoleDescriptorType">
                <sequence>
                    <element ref="md:AuthnQueryService" maxOccurs="unbounded"/>
                    <element ref="md:AssertionIDRequestService" minOccurs="0"
maxOccurs="unbounded"/>
                    <element ref="md:NameIDFormat" minOccurs="0"
maxOccurs="unbounded"/>
                </sequence>
            </extension>
        </complexContent>
    </complexType>
    <element name="AuthnQueryService" type="md:EndpointType"/>

    <element name="PDPDescriptor" type="md:PDPDescriptorType"/>
    <complexType name="PDPDescriptorType">
        <complexContent>
            <extension base="md:RoleDescriptorType">
                <sequence>
                    <element ref="md:AuthzService" maxOccurs="unbounded"/>
                    <element ref="md:AssertionIDRequestService" minOccurs="0"
maxOccurs="unbounded"/>
                    <element ref="md:NameIDFormat" minOccurs="0"
maxOccurs="unbounded"/>
                </sequence>
            </extension>
        </complexContent>
    </complexType>
    <element name="AuthzService" type="md:EndpointType"/>

    <element name="AttributeAuthorityDescriptor"
type="md:AttributeAuthorityDescriptorType"/>
    <complexType name="AttributeAuthorityDescriptorType">
        <complexContent>
            <extension base="md:RoleDescriptorType">
                <sequence>
                    <element ref="md:AttributeService" maxOccurs="unbounded"/>
                    <element ref="md:AssertionIDRequestService" minOccurs="0"
maxOccurs="unbounded"/>
                    <element ref="md:NameIDFormat" minOccurs="0"
maxOccurs="unbounded"/>
                    <element ref="md:AttributeProfile" minOccurs="0"
maxOccurs="unbounded"/>
                    <element ref="saml:Attribute" minOccurs="0"
maxOccurs="unbounded"/>
```

```
                </sequence>
            </extension>
        </complexContent>
    </complexType>
    <element name="AttributeService" type="md:EndpointType"/>

    <element name="AffiliationDescriptor" type="md:AffiliationDescriptorType"/>
    <complexType name="AffiliationDescriptorType">
        <sequence>
            <element ref="ds:Signature" minOccurs="0"/>
            <element ref="md:Extensions" minOccurs="0"/>
            <element ref="md:AffiliateMember" maxOccurs="unbounded"/>
            <element ref="md:KeyDescriptor" minOccurs="0" maxOccurs="unbounded"/>
        </sequence>
        <attribute name="affiliationOwnerID" type="md:entityIDType"
use="required"/>
        <attribute name="validUntil" type="dateTime" use="optional"/>
        <attribute name="cacheDuration" type="duration" use="optional"/>
        <attribute name="ID" type="ID" use="optional"/>
        <anyAttribute namespace="##other" processContents="lax"/>
    </complexType>
    <element name="AffiliateMember" type="md:entityIDType"/>
</schema>
```

A.29 SAML Schema protocol

C'est la liste pour le schéma du protocole SAML.

```
<?xml version="1.0" encoding="UTF-8"?>
<schema
    targetNamespace="urn:oasis:names:tc:SAML:2.0:protocol"
    xmlns="http://www.w3.org/2001/XMLSchema"
    xmlns:samlp="urn:oasis:names:tc:SAML:2.0:protocol"
    xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion"
    xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
    elementFormDefault="unqualified"
    attributeFormDefault="unqualified"
    blockDefault="substitution"
    version="2.0">
    <import namespace="urn:oasis:names:tc:SAML:2.0:assertion"
        schemaLocation="saml-schema-assertion-2.0.xsd"/>
    <import namespace="http://www.w3.org/2000/09/xmldsig#"
        schemaLocation="http://www.w3.org/TR/2002/REC-xmldsig-core-
20020212/xmldsig-core-schema.xsd"/>
    <annotation>
        <documentation>
            Document identifier: saml-schema-protocol-2.0
            Location: http://docs.oasis-open.org/security/saml/v2.0/
            Revision history:
            V1.0 (November, 2002):
              Initial Standard_u83 schema.
            V1.1 (September, 2003):
              Updates within the same V1.0 namespace.
            V2.0 (March, 2005):
              New protocol schema based in a SAML V2.0 namespace.
     </documentation>
    </annotation>
    <complexType name="RequestAbstractType" abstract="true">
        <sequence>
            <element ref="saml:Issuer" minOccurs="0"/>
            <element ref="ds:Signature" minOccurs="0"/>
            <element ref="samlp:Extensions" minOccurs="0"/>
        </sequence>
        <attribute name="ID" type="ID" use="required"/>
        <attribute name="Version" type="string" use="required"/>
        <attribute name="IssueInstant" type="dateTime" use="required"/>
        <attribute name="Destination" type="anyURI" use="optional"/>
         <attribute name="Consent" type="anyURI" use="optional"/>
    </complexType>
```

```
    <element name="Extensions" type="samlp:ExtensionsType"/>
    <complexType name="ExtensionsType">
        <sequence>
            <any namespace="##other" processContents="lax"
maxOccurs="unbounded"/>
        </sequence>
    </complexType>
    <complexType name="StatusResponseType">
         <sequence>
            <element ref="saml:Issuer" minOccurs="0"/>
            <element ref="ds:Signature" minOccurs="0"/>
            <element ref="samlp:Extensions" minOccurs="0"/>
            <element ref="samlp:Status"/>
         </sequence>
         <attribute name="ID" type="ID" use="required"/>
         <attribute name="InResponseTo" type="NCName" use="optional"/>
         <attribute name="Version" type="string" use="required"/>
         <attribute name="IssueInstant" type="dateTime" use="required"/>
         <attribute name="Destination" type="anyURI" use="optional"/>
         <attribute name="Consent" type="anyURI" use="optional"/>
    </complexType>
    <element name="Status" type="samlp:StatusType"/>
    <complexType name="StatusType">
        <sequence>
            <element ref="samlp:StatusCode"/>
            <element ref="samlp:StatusMessage" minOccurs="0"/>
            <element ref="samlp:StatusDetail" minOccurs="0"/>
        </sequence>
    </complexType>
    <element name="StatusCode" type="samlp:StatusCodeType"/>
    <complexType name="StatusCodeType">
        <sequence>
            <element ref="samlp:StatusCode" minOccurs="0"/>
        </sequence>
        <attribute name="Value" type="anyURI" use="required"/>
    </complexType>
    <element name="StatusMessage" type="string"/>
    <element name="StatusDetail" type="samlp:StatusDetailType"/>
    <complexType name="StatusDetailType">
        <sequence>
            <any namespace="##any" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
        </sequence>
    </complexType>
    <element name="AssertionIDRequest" type="samlp:AssertionIDRequestType"/>
    <complexType name="AssertionIDRequestType">
         <complexContent>
            <extension base="samlp:RequestAbstractType">
                <sequence>
                    <element ref="saml:AssertionIDRef" maxOccurs="unbounded"/>
                </sequence>
            </extension>
         </complexContent>
    </complexType>
    <element name="SubjectQuery" type="samlp:SubjectQueryAbstractType"/>
    <complexType name="SubjectQueryAbstractType" abstract="true">
         <complexContent>
            <extension base="samlp:RequestAbstractType">
                <sequence>
                    <element ref="saml:Subject"/>
                </sequence>
            </extension>
         </complexContent>
    </complexType>
    <element name="AuthnQuery" type="samlp:AuthnQueryType"/>
    <complexType name="AuthnQueryType">
        <complexContent>
            <extension base="samlp:SubjectQueryAbstractType">
                <sequence>
                    <element ref="samlp:RequestedAuthnContext" minOccurs="0"/>
                </sequence>
```

```
                <attribute name="SessionIndex" type="string" use="optional"/>
            </extension>
        </complexContent>
    </complexType>
    <element name="RequestedAuthnContext"
type="samlp:RequestedAuthnContextType"/>
    <complexType name="RequestedAuthnContextType">
        <choice>
            <element ref="saml:AuthnContextClassRef" maxOccurs="unbounded"/>
            <element ref="saml:AuthnContextDeclRef" maxOccurs="unbounded"/>
        </choice>
        <attribute name="Comparison" type="samlp:AuthnContextComparisonType"
use="optional"/>
    </complexType>
    <simpleType name="AuthnContextComparisonType">
        <restriction base="string">
            <enumeration value="exact"/>
            <enumeration value="minimum"/>
            <enumeration value="maximum"/>
            <enumeration value="better"/>
        </restriction>
    </simpleType>
    <element name="AttributeQuery" type="samlp:AttributeQueryType"/>
    <complexType name="AttributeQueryType">
        <complexContent>
            <extension base="samlp:SubjectQueryAbstractType">
                <sequence>
                    <element ref="saml:Attribute" minOccurs="0"
maxOccurs="unbounded"/>
                </sequence>
            </extension>
        </complexContent>
    </complexType>
    <element name="AuthzDecisionQuery" type="samlp:AuthzDecisionQueryType"/>
    <complexType name="AuthzDecisionQueryType">
        <complexContent>
            <extension base="samlp:SubjectQueryAbstractType">
                <sequence>
                    <element ref="saml:Action" maxOccurs="unbounded"/>
                    <element ref="saml:Evidence" minOccurs="0"/>
                </sequence>
                <attribute name="Resource" type="anyURI" use="required"/>
            </extension>
        </complexContent>
    </complexType>
    <element name="AuthnRequest" type="samlp:AuthnRequestType"/>
    <complexType name="AuthnRequestType">
        <complexContent>
            <extension base="samlp:RequestAbstractType">
                <sequence>
                    <element ref="saml:Subject" minOccurs="0"/>
                    <element ref="samlp:NameIDPolicy" minOccurs="0"/>
                    <element ref="saml:Conditions" minOccurs="0"/>
                    <element ref="samlp:RequestedAuthnContext" minOccurs="0"/>
                    <element ref="samlp:Scoping" minOccurs="0"/>
                </sequence>
                <attribute name="ForceAuthn" type="boolean" use="optional"/>
                <attribute name="IsPassive" type="boolean" use="optional"/>
                <attribute name="ProtocolBinding" type="anyURI" use="optional"/>
                <attribute name="AssertionConsumerServiceIndex"
type="unsignedShort" use="optional"/>
                <attribute name="AssertionConsumerServiceURL" type="anyURI"
use="optional"/>
                <attribute name="AttributeConsumingServiceIndex"
type="unsignedShort" use="optional"/>
                <attribute name="ProviderName" type="string" use="optional"/>
            </extension>
        </complexContent>
    </complexType>
    <element name="NameIDPolicy" type="samlp:NameIDPolicyType"/>
    <complexType name="NameIDPolicyType">
```

```
        <attribute name="Format" type="anyURI" use="optional"/>
        <attribute name="SPNameQualifier" type="string" use="optional"/>
        <attribute name="AllowCreate" type="boolean" use="optional"/>
    </complexType>
    <element name="Scoping" type="samlp:ScopingType"/>
    <complexType name="ScopingType">
        <sequence>
            <element ref="samlp:IDPList" minOccurs="0"/>
            <element ref="samlp:RequesterID" minOccurs="0"
maxOccurs="unbounded"/>
        </sequence>
        <attribute name="ProxyCount" type="nonNegativeInteger" use="optional"/>
    </complexType>
    <element name="RequesterID" type="anyURI"/>
    <element name="IDPList" type="samlp:IDPListType"/>
    <complexType name="IDPListType">
        <sequence>
            <element ref="samlp:IDPEntry" maxOccurs="unbounded"/>
            <element ref="samlp:GetComplete" minOccurs="0"/>
        </sequence>
    </complexType>
    <element name="IDPEntry" type="samlp:IDPEntryType"/>
    <complexType name="IDPEntryType">
        <attribute name="ProviderID" type="anyURI" use="required"/>
        <attribute name="Name" type="string" use="optional"/>
        <attribute name="Loc" type="anyURI" use="optional"/>
    </complexType>
    <element name="GetComplete" type="anyURI"/>
    <element name="Response" type="samlp:ResponseType"/>
    <complexType name="ResponseType">
         <complexContent>
            <extension base="samlp:StatusResponseType">
                <choice minOccurs="0" maxOccurs="unbounded">
                    <element ref="saml:Assertion"/>
                    <element ref="saml:EncryptedAssertion"/>
                </choice>
            </extension>
         </complexContent>
    </complexType>
    <element name="ArtifactResolve" type="samlp:ArtifactResolveType"/>
    <complexType name="ArtifactResolveType">
         <complexContent>
            <extension base="samlp:RequestAbstractType">
                <sequence>
                    <element ref="samlp:Artifact"/>
                </sequence>
            </extension>
         </complexContent>
    </complexType>
    <element name="Artifact" type="string"/>
    <element name="ArtifactResponse" type="samlp:ArtifactResponseType"/>
    <complexType name="ArtifactResponseType">
         <complexContent>
            <extension base="samlp:StatusResponseType">
                <sequence>
                    <any namespace="##any" processContents="lax" minOccurs="0"/>
                </sequence>
            </extension>
         </complexContent>
    </complexType>
    <element name="ManageNameIDRequest" type="samlp:ManageNameIDRequestType"/>
    <complexType name="ManageNameIDRequestType">
         <complexContent>
            <extension base="samlp:RequestAbstractType">
                <sequence>
                    <choice>
                        <element ref="saml:NameID"/>
                        <element ref="saml:EncryptedID"/>
                    </choice>
                    <choice>
                        <element ref="samlp:NewID"/>
```

```
                        <element ref="samlp:NewEncryptedID"/>
                        <element ref="samlp:Terminate"/>
                    </choice>
                </sequence>
            </extension>
        </complexContent>
    </complexType>
    <element name="NewID" type="string"/>
    <element name="NewEncryptedID" type="saml:EncryptedElementType"/>
    <element name="Terminate" type="samlp:TerminateType"/>
    <complexType name="TerminateType"/>
    <element name="ManageNameIDResponse" type="samlp:StatusResponseType"/>
    <element name="LogoutRequest" type="samlp:LogoutRequestType"/>
    <complexType name="LogoutRequestType">
        <complexContent>
            <extension base="samlp:RequestAbstractType">
                <sequence>
                    <choice>
                        <element ref="saml:BaseID"/>
                        <element ref="saml:NameID"/>
                        <element ref="saml:EncryptedID"/>
                    </choice>
                    <element ref="samlp:SessionIndex" minOccurs="0"
maxOccurs="unbounded"/>
                </sequence>
                <attribute name="Reason" type="string" use="optional"/>
                <attribute name="NotOnOrAfter" type="dateTime" use="optional"/>
            </extension>
        </complexContent>
    </complexType>
    <element name="SessionIndex" type="string"/>
    <element name="LogoutResponse" type="samlp:StatusResponseType"/>
    <element name="NameIDMappingRequest" type="samlp:NameIDMappingRequestType"/>
    <complexType name="NameIDMappingRequestType">
        <complexContent>
            <extension base="samlp:RequestAbstractType">
                <sequence>
                    <choice>
                        <element ref="saml:BaseID"/>
                        <element ref="saml:NameID"/>
                        <element ref="saml:EncryptedID"/>
                    </choice>
                    <element ref="samlp:NameIDPolicy"/>
                </sequence>
            </extension>
        </complexContent>
    </complexType>
    <element name="NameIDMappingResponse"
type="samlp:NameIDMappingResponseType"/>
    <complexType name="NameIDMappingResponseType">
        <complexContent>
            <extension base="samlp:StatusResponseType">
                <choice>
                    <element ref="saml:NameID"/>
                    <element ref="saml:EncryptedID"/>
                </choice>
            </extension>
        </complexContent>
    </complexType>
</schema>
```

A.30 SAML Schema X.500

C'est la liste pour le SAML X.500.

```
<?xml version="1.0" encoding="UTF-8"?>
<schema
    targetNamespace="urn:oasis:names:tc:SAML:2.0:profiles:attribute:X500"
    xmlns="http://www.w3.org/2001/XMLSchema"
```

```
    elementFormDefault="unqualified"
    attributeFormDefault="unqualified"
    blockDefault="substitution"
    version="2.0">
    <annotation>
        <documentation>
            Document identifier: saml-schema-x500-2.0
            Location: http://docs.oasis-open.org/security/saml/v2.0/
            Revision history:
              V2.0 (March, 2005):
                Custom schema for X.500 attribute profile, first published in SAML 2.0.
        </documentation>
    </annotation>
    <attribute name="Encoding" type="string"/>
</schema>
```

A.31 Schéma SAML XACML

C'est la liste pour le SAML XACML.

```
<?xml version="1.0" encoding="UTF-8"?>
<schema
    targetNamespace="urn:oasis:names:tc:SAML:2.0:profiles:attribute:XACML"
    xmlns="http://www.w3.org/2001/XMLSchema"
    elementFormDefault="unqualified"
    attributeFormDefault="unqualified"
    blockDefault="substitution"
    version="2.0">
    <annotation>
        <documentation>
            Document identifier: saml-schema-xacml-2.0
            Location: http://docs.oasis-open.org/security/saml/v2.0/
            Revision history:
            V2.0 (March, 2005):
              Custom schema for XACML attribute profile, first published in SAML 2.0.
        </documentation>
    </annotation>
    <attribute name="DataType" type="anyURI"/>
</schema>
```

Appendice I

Considérations sur la sécurité et la confidentialité

Sécurité et confidentialité doivent être traitées de façon systémique, en considérant les facteurs humains tels que les conflits d'ingénierie sociale, les questions de politique, la gestion des clés et la gestion de la confiance, la sécurisation des implémentations et d'autres facteurs qui sortent du domaine d'application du présent appendice. Les solutions techniques de la sécurité ont un coût, et les exigences et les politiques de remplacement doivent aussi être prises en considération, comme doivent l'être les exigences légales et réglementaires.

Le présent appendice récapitule les questions et approches générales de la sécurité ainsi que les menaces spécifiques et leurs contre-mesures pour l'utilisation des assertions, protocoles, liaisons et profils SAML d'une façon sécurisée qui préserve la confidentialité. Le présent appendice décrit et analyse les propriété de SAML en ce qui concerne la sécurité et la confidentialité. L'intention est de fournir des informations aux architectes et développeurs de systèmes fondés sur SAML à propos de ce qui suit:

- les questions de confidentialité à considérer et comment l'architecture de SAML traite ces questions;
- les menaces, et donc les risques pour la sécurité, auxquels sont soumis les systèmes fondés sur SAML;
- les risques pour la sécurité que l'architecture de SAML prend en compte, et comment elle le fait;
- les risques pour la sécurité qu'elle ne prend pas en compte;
- les recommandations pour des contre-mesures qui atténuent ces risques pour la sécurité.

I.1 Vie privée

SAML comporte la capacité de faire des déclarations sur les attributs et autorisations des entités authentifiées. Il y a de très nombreuses situations courantes où les informations portées dans ces déclarations sont des choses qu'une ou plusieurs des parties à une communication désirerait ne rendre accessibles qu'à un ensemble d'entités aussi restreint que possible. Les déclarations d'attributs médicaux ou financiers sont de simples exemples de tels cas.

De nombreux pays et juridictions ont des lois et réglements concernant la vie privée qui devraient être prises en considération lors du développement de systèmes fondés sur SAML. Les parties qui font des déclarations, produisent des assertions, convoient des assertions, et consomment des assertions doivent être au courant de ces problèmes potentiels de vie privée et devraient essayer de les régler dans leurs mises en œuvre de systèmes fondés sur SAML.

I.2 Confidentialité

Peut-être l'aspect le plus important pour assurer la confidentialité aux parties à une transaction fondée sur SAML est la capacité à mener à bien la transaction avec une garantie de confidentialité. En d'autres termes, les informations contenues dans une assertion peuvent-elles, et seulement elles, être convoyées du producteur à l'audience de destination sans les rendre accessibles à d'autres parties?

Il est techniquement possible de convoyer des informations de façon confidentielle. Toutes les parties aux transactions fondées sur SAML devraient analyser chaque étape de l'interaction (et toute utilisation ultérieure des données obtenues de ces transactions) pour s'assurer que les informations qui devraient être gardées confidentielle le sont réellement.

On devrait aussi noter que simplement obscurcir le contenu des assertions peut n'être pas une protection adéquate de la confidentialité. Il y a de nombreux cas où la simple disponibilité de l'information qu'un utilisateur donné (ou une adresse IP) a accédé à un service donné, peut constituer une atteinte à la confidentialité (par exemple, l'information qu'un usager accède à un service de consultation médicale pour une assertion peut être une atteinte à la vie privée, sans connaître le contenu de l'assertion). Des solutions partielles à ces problèmes peuvent être fournies par diverses techniques d'interaction anonyme, comme décrit dans les prochains paragraphes.

I.3 Pseudonyme et anonymat

Il n'y a pas de définition de l'anonymat qui soit satisfaisante pour tous les cas. De nombreuses définitions traitent le simple cas d'un envoyeur et d'un message, et discutent de "l'anonymat" sous l'aspect de ne pouvoir lier un envoyeur donné à un message envoyé, ou du renvoi d'un message à l'envoyeur. Alors que cette définition est adéquate pour le cas exceptionnel, elle ignore l'agrégation d'informations qui est possible à la longue sur la base du comportement plutôt que sur un identifiant.

Dans SAML, il est utile de penser l'anonymat comme étant "dans un ensemble". Cette notion est pertinente pour SAML à cause de l'utilisation d'autorités. Même si un Sujet est "anonyme", ce sujet est encore identifiable comme membre de l'ensemble des Sujets au sein du domaine de l'autorité compétente. Les sytèmes qui ont la capacité SAML sont limités à un "anonymat partiel" au mieux, à cause de l'utilisation des autorités. Une entité sur laquelle est faite une assertion est identifiable comme une des entités du groupe qui est en relation avec l'autorité productrice.

Les limitations à l'anonymat peuvent être plus strictes qu'une simple association d'autorités, en fonction de la façon dont les identifiants sont utilisés, car la réutilisation de pseudonymes identifiants permet l'émergence d'informations d'identification potentielles. De plus, les usagers de systèmes à capacité SAML peuvent aussi rendre l'atteinte à l'anonymat pire par leurs actions.

A part l'identité légale, tout identifiant pour un sujet peut être considéré comme un pseudonyme. Et même des notions comme celle de "détenteur de clé" peuvent être considérées comme servant d'équivalent à un pseudonyme pour lier une action (ou ensemble d'actions) à un sujet. Même une description telle que "l'utilisateur vient de demander l'accès à l'objet XYZ à 23:34" peut servir d'équivalent d'un pseudonyme.

Et donc, par rapport à la "capacité à nuire," il ne fait aucune différence que l'utilisateur soit décrit avec un identifiant ou par un comportement (par exemple, utiliser une clé ou effectuer une action).

Ce qui fait la différence est le nombre de fois qu'un équivalent particulier d'un pseudonyme est utilisé. L'anonymat donne une taxonomie de pseudonymes commençant par des pseudonymes personnels (comme des surnoms) qui sont utilisés tout le temps, jusqu'à divers types de pseudonymes de rôle (comme Secrétaire à la Défense), et les pseudonymes "à utilisation unique".

Les pseudonymes à utilisation unique peuvent donner l'anonymat (dans SAML, considérer cela comme "anonymat dans un ensemble"). Cependant, plus un pseudonyme donné est utilisé souvent, plus grand est le risque pour l'anonymat. En d'autres termes, réutiliser un pseudonyme permet que des informations potentiellement identifiantes soient associées au pseudonyme. A la longue, cela conduira à une accumulation qui peut identifier de façon univoque l'identité associée à un pseudonyme.

Les autorités de site d'origine (telles que les autorités d'authentification et les autorités d'attribut) peuvent fournir un degré "d'anonymat partiel" en employant des identifiants ou clés à usage unique (pour le cas de "détenteur de clé"). Cet anonymat est "partiel" au mieux parce que le Sujet est nécessairement confiné à l'ensemble des sujets en relation avec l'autorité. Cet ensemble peut être encore réduit (réduisant encore l'anonymat) lorsque des attributs agrégés sont utilisés qui subdivisent la communauté des utilisateurs au site d'origine. Les usagers qui sont réellement soucieux d'anonymat doivent veiller à déguiser ou éviter les schémas de comportement inhabituels qui pourraient servir à les "désanonymiser" à la longue.

I.4 Sécurité

Les paragraphes suivant exposent les considérations sur la sécurité.

I.4.1 Fondements

Les communications entre systèmes fondés sur des ordinateurs sont sujettes à diverses menaces, et ces menaces portent un certain niveau de risque associé. La nature du risque dépend d'une série de facteurs, incluant la nature des communications, la nature des systèmes communiquants, des supports de communication, de l'environnement de communication, des environments de systèmes terminaux, et ainsi de suite.

SAML est destiné à aider les développeurs à établir des contextes de sécurité pour des communications de niveau application fondées sur des ordinateurs au sein ou entre domaines de sécurité. Dans ce rôle, SAML transfère des données d'authentification, qui prennent en charge la capacité des systèmes terminaux à protéger contre les utilisations non autorisées. La sécurité des communications est directement applicable à la conception de SAML. La sécurité des systèmes est principalement visée dans le contexte des modèles de menaces de SAML.

I.4.2 Domaine d'application

Certains domaines qui impactent largement la sécurité globale d'un système qu'utilise SAML sont explicitement en dehors du domaine d'application de SAML. Alors que la présente Recommandation ne vise pas ces domaines, ils devraient toujours être pris en considération lors de la révision de la sécurité d'un système. En particulier, ces questions sont importantes, mais actuellement en dehors du domaine d'applicaton de SAML:

- authentification initiale: SAML permet de faire des déclarations sur des actes d'authentification qui sont survenus, mais n'inclut aucune exigence ou spécification pour ces actes d'authentification. Les consommateurs des assertions d'authentification devrait être prévenus contre une confiance aveugle à l'égard de ces assertions à moins qu'ils ne connaissent les bases qui ont servi à les établir. La confiance dans les assertions ne doit jamais excéder la confiance que le producteur d'assertions est arrivé de façon correcte aux conclusions attestées;

- modèle de confiance: dans de nombreux cas, la sécurité d'une conversation SAML va dépendre du modèle de confiance sous-jacent, qui est normalement fondé sur une infrastructure de gestion de clés (par exemple, PKI ou une clé secrète). Par exemple, les messages SOAP sécurisés au moyen d'une signature XML ne sont sécurisés que pour autant que les clés utilisées dans l'échange puissent être de confiance. Des clés compromises non détectées, ou des certificats révoqués, par exemple, pourraient permettre des atteintes à la sécurité. Même l'échec à exiger un certificat ouvre la porte à des attaques par imitation. L'établissement de PKI n'est pas trivial et doit être implémenté correctement afin de sécuriser les couches qui sont construites par-dessus (comme des parties de SAML).

Une implémentation convenable des protocoles de sécurité est nécessaire pour maintenir la sécurité d'un système, y compris la génération de nombres aléatoires ou pseudo aléatoires et la mise en mémoire sécurisée des clés.

I.4.3 Modèle de menace SAML

Le modèle général des menaces de l'Internet décrit dans les lignes directrices de l'IETF sur les considérations de sécurité est la base du modèle de menaces de SAML. On suppose ici que les deux, ou plus, points d'extrémité d'une transaction SAML ne sont pas compromis, mais que l'attaquant a l'entier contrôle du canal de communications.

De plus, du fait de la nature de SAML comme protocole multi-partie de déclarations d'authentification et d'autorisation, on doit considérer les cas où une ou plusieurs des parties à une transaction SAML légitime – qui opèrent légitimement dans leur rôle pour cette transaction—essayent d'utiliser de façon malveillante dans une transaction ultérieure les informations obtenues dans cette transaction.

Les scénarios suivants décrivent les attaques possibles:

- **collusion**: coopération secrète entre deux ou plusieurs entités système pour lancer une attaque, par exemple:
  - collusion entre principal et fournisseur de service;
  - collusion entre principal et fournisseur d'identité;
  - collusion entre fournisseur d'identité et fournisseur de service;
  - collusion parmi deux principaux ou plus;
  - collusion entre deux fournisseurs de service ou plus;
  - collusion entre deux fournisseurs d'identité ou plus;
- **attaque de déni de service**: empêcher l'accès autorisé à une ressource système ou retarder les opérations et fonctions du système;
- **attaque par intrusion**: forme d'attaque par espionnage actif dans laquelle l'attaquent intercepte et modifie de façon sélective les données communiquées et se déguise en une ou plusieurs des entités impliquées dans une association de communication;
- **attaque en répétition**: attaque dans laquelle une transmission de données valide est répétée de façon malveillante ou frauduleuse, soit par l'entité d'origine, soit par un adversaire qui intercepte les données et les retransmet, éventuellement au titre d'une attaque sous un déguisement;
- **capture de session**: forme d'espionnage actif dans laquelle l'attaquant s'empare du contrôle d'une association de communication établie antérieurement.

Dans tous les cas, les mécanismes locaux qu'utiliseront les systèmes pour décider de générer ou non des assertions sont en dehors du domaine d'application de ce document. Et donc, les menaces qui proviennent des détails de l'ouverture de session originale chez une autorité d'authentification, par exemple, sont également en dehors de ce domaine. Si une autorité produit une fausse assertion, les menaces provenant de la consommation de cette assertion dans les systèmes en aval sont donc explicitement en dehors du domaine d'application.

La conséquence directe d'une telle délimitation de domaine est que la sécurité d'un système fondé sur l'entrée d'assertions n'est qu'aussi bonne que celle du système utilisé pour générer ces assertions, et de la rectitude des données et traitements sur lesquels les assertions générées se fondent. Lorsqu'on veut déterminer à quels producteurs faire confiance, en particulier dans les cas où les assertions seront utilisées comme entrées pour l'authentification ou des décisions d'autorisation, le risque de compromettre la sécurité qui résulte de la consommation d'assertions fausses mais dont la fourniture est valide, est important. Les politiques de confiance entre producteurs et consommateurs d'assertions devraient toujours être écrites pour inclure des considérations de responsabilité significatives et les mises en œuvre devraient fournir des moyens d'audit appropriés.

I.5 Techniques de sécurité

Les paragraphes suivants décrivent les techniques de sécurité et diverses technologies disponibles à implémenter dans les développements SAML.

I.5.1 Authentification

Ici, authentification signifie la capacité d'une partie à une transaction à déterminer l'identité de l'autre partie à la transaction. Cette authentification peut être dans une seule direction ou elle peut être bilatérale.

- **Session active**: une authentification durable est fournie par le canal de communications utilisé pour transporter un message SAML. Cette authentification peut être unilatérale – de l'initiateur de la session au receveur – ou bilatérale. La méthode spécifique sera déterminée par le protocole de communications utilisé. Par exemple, l'utilisation d'un protocole réseau sécurisé, tel que TLS ou le protocole de sécurité IP, fournit à l'expéditeur de message SAML la capacité à authentifier la destination pour l'environnement TCP/IP.
- **Niveau de message**: XML Signature du W3C et OASIS WSS fournissent des méthodes de création d'une "authentification" durable qui est étroitement couplée à un document. Cette méthode ne garantit pas de façon indépendante que l'expéditeur du message soit en fait le signataire (et bien sûr, dans de nombreux cas où des intermédiaires sont impliqués, cela n'est explicitement pas le cas). Toute méthode qui permet la confirmation persistante de l'implication d'une entité résoluble de façon univoque avec un sous ensemble donné d'un message XML est suffisante pour satisfaire cette exigence.

I.5.2 Confidentialité

Confidentialité signifie que le contenu d'un message ne peut être lu que par les receveurs souhaités et par personne d'autre qui rencontrerait le message.

- **En transit**: utilisation d'un protocole réseau sécurisé tel que TLS ou le protocole de sécurité IP qui fournit la confidentialité transitoire d'un message alors qu'il est transféré entre deux nœuds.
- **Niveau de message:** le chiffrement XML fournit le chiffrement sélectif des documents XML. Cette méthode de chiffrement fournit une confidentialité persistante et sélective des éléments au sein d'un message XML.

I.5.3 Intégrité des données

L'intégrité des données est la capacité à confirmer qu'un message donné est reçu sans altération d'après la version du message envoyé.

- **En transit**: utilisation d'un protocole réseau sécurisé tel que TLS ou le protocole de sécurité IP qui peut être configuré pour fournir la protection de l'intégrité pour les paquets transmis via la connexion réseau.
- **Niveau de message**: XML Signature fournit une méthode de création d'une garantie durable de l'absence d'altération d'un message qui est étroitement couplée à ce message. Toute méthode qui permet la confirmation persistante de l'implication d'une entité résoluble de façon univoque avec un sous ensemble donné d'un message XML est suffisante pour satisfaire cette exigence.

I.5.4 Notes sur la gestion de clé

De nombreux points du présent appendice se réfèrent à la capacité des systèmes à fournir l'authentification, l'intégrité des données, et la confidentialité via divers schémas impliquant la signature numérique et le chiffrement. Pour tous ces schémas, la sécurité fournie par le schéma est limitée en fonction des systèmes de gestion de clé en place. Certaines limitations spécifiques sont précisées ci-dessous.

1) **Accès à la clé**: on supppose que, si les systèmes fondés sur des clés sont utilisés pour l'authentification, l'intégrité des données, et la non-répudiation, la sécurité est mise en place pour garantir que l'accès à une clé privée ou secrète représentant un principal n'est pas disponible à des parties inappropriées. Par exemple, une signature numérique créée avec la clé privée de Bob est seulement la preuve de l'implication de Bob dans la mesure où Bob est le seul qui ait accès à la clé. En général, l'accès aux clés devrait être limité à l'ensemble minimum d'entités possible (particulièrement important pour les clés d'entreprise ou d'organisations) et devrait être protégé par des phrases de code et autres moyens. On doit appliquer des précautions de sécurité standard (ne pas écrire la phrase de code, ne pas laisser une fenêtre ouverte avec l'accès par la clé sur l'ordinateur quand on est absent, et ainsi de suite).
2) **Liaison de l'identité à la clé**: pour un système à clé utilisé pour l'authentification, il doit y avoir un lien de confiance de l'identité à la clé. Vérifier une signature numérique sur un document peut déterminer si le document n'est pas altéré depuis sa signature, et s'il a été réellement signé par une clé donnée. Cependant, cela ne confirme pas que la clé utilisée est réellement la clé d'un individu spécifique approprié à l'instant et à l'objet. Vérifier la liaison d'une clé à une partie exige une validation supplémentaire.

Cette liaison entre clé et individu doit être établie. Les solutions courantes incluent des répertoires locaux qui mémorisent à la fois identifiants et clés – ce qui est simple à comprendre mais difficile à entretenir – ou l'utilisation de certificats. Utiliser des certificats peut fournir un moyen mesurable d'associer une clé à une identité, mais requiert des mécanismes pour gérer la durée de vie des certificats et les changements du statut de la liaison (par exemple, un employé quitte l'entreprise et n'a plus le statut d'employé). Une approche courante est d'utiliser une infrastructure de clé publique (PKI, *public key infrastructure*).

Dans ce cas, un ensemble d'autorités de certificat (CA, *certifying authority*) racine de confiance est identifié pour chaque consommateur de signatures – répondant à la question "En qui ai-je confiance pour faire des déclarations de liaison d'identité à clé ?" La vérification d'une signature devient alors un processus qui consite d'abord à vérifier la signature (pour déterminer si la signature a été faite par la clé en question et si le message n'a pas été changé) et puis à valider la chaîne de certificat (pour déterminer si la clé est liée à la bonne identité) et à valider que la liaison est toujours appropriée. Valider la liaison exige de franchir des étapes pour s'assurer que la liaison est en cours de validité – un certificat a normalement une "durée de vie" incorporée, mais si une clé est compromise durant la vie d'un certificat, la liaison clé à identité contenue dans le certificat devient non valide alors que le certificat apparaît toujours valide. Aussi, les certificats dépendent souvent des associations qui peuvent se terminer avant l'expiration de leur durée de vie (par exemple, les certificats qui devraient devenir non valides quand quelqu'un change d'employeur, etc.). Un système approprié de gestion de clé est donc assez fort mais très complexe. Vérifier une signature finit par être un processus de vérification de la liaison document à clé, puis de vérification de la liaison clé à identité, autant que de la validité en cours de la clé et du certificat.

I.5.5 Suites de chiffrement TLS

L'utilisation de HTTP sur SSL 3.0 (voir l'Appendice IV) ou de TLS 1.0, ou l'utilisation d'URL avec le schéma d'URL HTTPS, est fortement recommandée à de nombreux endroits dans la présente Recommandation.

Sauf mention contraire, dans toute utilisation de SSL 3.0 ou TLS 1.0 par une liaison SAML, les serveurs doivent s'authentifier auprès des clients en utilisant un certificat X.509 v3. Le client doit établir l'identité du serveur sur la base du contenu du certificat (normalement, par examen du champ DN de sujet du certificat).

SSL/TLS peut être configuré pour utiliser de nombreuses suites de chiffrement différentes, dont toutes ne sont pas adéquates pour fournir les "meilleures pratiques" de sécurité. Une suite de chiffrement combine quatre sortes de caractéristiques de sécurité, et reçoit un nom en [SSL]. Avant que les données ne s'écoulent sur une connexion SSL, les deux extrémités essayent de négocier une suite de chiffrement. Ceci leur laisse établir une qualité de protection appropriée pour leurs communications, dans les contraintes des combinaisons de mécanismes particuliers disponibles. Les caractéristiques associées à une suite de chiffrement sont:

SSL définit de nombreux algorithmes d'échange de clés. Certains des mécanismes servent à l'authentification du serveur. Cependant, des mécanismes anonymes d'échange de clé sont aussi pris en charge. (Les algorithmes d'échange de clé anonyme sont sujets à des attaques "d'intrusion", et ne sont pas recommandés dans le contexte SAML). L'algorithme "RSA" d'échange de clé authentifié est actuellement l'algorithme le plus interopérable (la licence de l'algorithme RSA est arrivée à expiration). Un autre algorithme d'échange de clé important est l'échange de clé authentifié Diffie-Hellman "DHE_DSS", qui n'a pas de contrainte d'implémentation liée à la licence.

Savoir si l'algorithme d'échange de clé peut être librement exporté des États-Unis d'Amérique. Les algorithmes exportables doivent utiliser des clés publiques courtes (512 bits) pour l'échange de clés et des clés symétriques courtes (40 bits) pour le chiffrement. Des clés de cette longueur ont été attaquées avec succès, et leur utilisation n'est pas recommandée.

L'option d'algorithme de chiffrement la plus rapide est le flux de chiffrement RC4; DES et ses variantes (DES40, 3DES-EDE) ainsi que AES sont aussi acceptés en mode "chaînage de bloc de chiffrement" (CBC, *cipher block chaining*). D'autres modes sont aussi acceptés; se reporter à la documentation TLS.

Le chiffrement nul est une option dans certaines suites de chiffrement. Le chiffrement nul n'effectue aucun chiffrement; dans de tels cas, SSL/TLS n'est utilisé que pour authentifier et fournir la protection d'intégrité. Les suites de chiffrement avec le chiffrement nul ne fournissent pas la confidentialité, et ne doivent pas être utilisées dans les cas où la confidentialité est une exigence et n'est pas obtenue par d'autre moyen que SSL/TLS.

L'algorithme digest *(résumé)* est utilisé pour le code d'authentification de message. La FCC a récemment recommandé d'utiliser SHA-256 et l'IETF a décidé de suivre.

I.6 Généralités sur la sécurité dans SAML

Les paragraphes suivants analysent les risques pour la sécurité en utilisant et en implémentant SAML, et ils décrivent les contre-mesures pour diminuer les risques.

I.6.1 Assertions SAML

Au niveau de l'assertion SAML elle-même, il y a peu à dire sur les questions de sécurité — la plupart des questions surviennent au cours des communications dans le protocole de demande/réponse, ou durant les tentatives d'utiliser SAML au moyen d'une des liaisons. Le consommateur est, bien sûr, toujours supposé respecter l'intervalle de validité de l'assertion et tout élément `<OneTimeUse>` présent dans l'assertion.

Cependant, une question mérite l'analyse au niveau de l'assertion: une assertion, une fois produite, est hors du contrôle du producteur. Ce fait a un certain nombre de ramifications. Par exemple, le producteur n'a pas le contrôle de la durée pendant laquelle l'assertion va persister dans les systèmes du consommateur; pas plus que le producteur n'a de contrôle sur les parties avec lesquelles le consommateur va partager les informations de l'assertion. Ces questions s'ajoutent à celles sur l'attaquant malveillant qui peut voir le contenu des assertions qui passent non chiffrées (ou insuffisamment chiffrées) sur le réseau.

Alors que des efforts ont été faits pour résoudre beaucoup de ces problèmes au sein de la Recommandation SAML, rien dans la présente Recommandation ne supprime l'exigence de prise en considération attentive de ce qu'on doit mettre dans une assertion. A tout moment, les producteurs devraient considérer les conséquences possibles du stockage des informations de l'assertion sur un site distant, où elles peuvent être directement utilisées à mauvais escient, ou exposées à un fouineur potentiel, ou peut-être mémorisées pour une utilisation frauduleuse plus créative. Les producteurs devraient aussi considérer la possibilité que les informations de l'assertion soient partagées avec d'autres parties, ou même rendues publiques, intentionnellement ou par inadvertence.

I.6.2 Protocole SAML

Le présent paragraphe décrit les considérations de sécurité pour le protocole SAML de demande-réponse lui-même, au-delà de toute menace résultant de l'utilisation d'une liaison de protocole particulière.

– Déni de service

  Le protocole SAML est susceptible d'une attaque de déni de service (DoS). Le traitement d'une demande SAML est potentiellement une opération très coûteuse, qui inclut l'analyse du message de demande (impliquant normalement la construction d'une arborescence DOM), des recherches de mémorisation de base de données/d'assertion (qui peuvent être non indexées), la construction d'un message de réponse, et éventuellement une ou plusieurs opérations de signature numérique. Et donc, les efforts requis d'un attaquant qui génère une demande sont bien inférieurs à l'effort nécessaire pour traiter cette demande.

1) Exiger l'authentification du client à un niveau inférieur

Exiger des clients qu'ils s'authentifient à un niveau inférieur au niveau du protocole SAML (par exemple, en utilisant SOAP sur une liaison HTTP, avec HTTP sur TLS/SSL, et avec une exigence que les certificats côté client aient une autorité de certificat de confiance à leur racine) procure la traçabilité dans le cas d'attaque de DoS.

Si l'authentification n'est utilisée que pour procurer la traçabilité, cela n'empêche pas en soi que survienne l'attaque, mais cela a un effet dissuasif.

Si l'authentification est couplée avec un système de contrôle d'accès, les attaques de DoS de la part d'agents externes sont effectivement bloquées. (Il est possible que surcharger le schéma d'authentification client puisse toujours fonctionner comme une attaque de déni de service sur le service SAML, mais cette attaque devra être traitée dans le contexte du schéma d'authentification client choisi.)

Quel que soit le système d'authentification client utilisé, il devrait procurer la capacité à résoudre une origine unique pour chaque demande, et ne devrait pas être sujet à falsification. (Par exemple, dans le cas de la traçabilité seule, charger l'adresse IP est insuffisant car cette information peut aisément être parodiée.)

2) Exiger une demande signée

Exiger une demande signée diminue aussi l'ordre de grandeur de l'asymétrie entre le travail fait par le demandeur et par le répondant. Le travail additionnel requis du répondant pour vérifier la signature est un pourcentage relativement faible de celui du travail total requis du répondant, alors que le processus de calcul de la signature numérique représente une quantité relativement importante de travail pour le demandeur. Réduire cette asymétrie diminue le risque associé aux attaques de DoS.

Cependant, un attaquant peut théoriquement capturer un message signé et le repasser continuellement, contournant ainsi cette exigence. Cette situation peut être évitée en exigeant l'utilisation de l'élément de signature XML Signature `<ds:SignatureProperties>` qui contient un horodatage; l'horodatage peut alors être utilisé pour déterminer si la signature est récente. Dans ce cas, plus la fenêtre temporelle pendant laquelle la signature est traitée comme valide après sa production est réduite, plus la sécurité contre les attaques de déni de service en répétition est élevée.

3) Restriction de l'accès à l'URL d'interaction

Limiter à un très faible niveau la capacité à produire une demande à un service SAML par un ensemble de parties connues réduit de façon drastique le risque d'attaque de DoS. Dans ce cas, seules les attaques originaires de l'ensemble fini de parties connues sont possibles, diminuant considérablement l'exposition, à la fois aux clients malveillants et aux attaques de DoS utilisant des machines compromises comme zombies.

Il y a de nombreuses méthodes possibles pour limiter l'accès, comme de placer le répondant SAML à l'intérieur d'un intranet sécurisé et d'implémenter des règles d'accès au niveau du routeur.

I.7 Considérations de sécurité sur les liaisons SAML

Les considérations de sécurité dans la conception du protocole de demande-réponse SAML dépendent dans une large mesure de la liaison de protocole particulière qui est utilisée. Les liaisons prises en charge sont la liaison SOAP, la liaison SOAP inverse (PAOS), la liaison HTTP Redirect, la liaison HTTP Redirect/POST, la liaison HTTP Artifact et les liaisons d'URI SAML.

I.7.1 Liaison SOAP SAML

Comme la liaison SOAP de SAML n'exige pas d'authentification et n'a pas d'exigence pour la confidentialité en transit ou l'intégrité de message, elle est offerte à une grande variété d'attaques courantes. Les considérations générales sont exposées séparément des considérations qui se rapportent au cas de SOAP sur HTTP.

1) Espionnage

Menace: comme il n'y a pas d'exigence de confidentialité en transit, il est possible qu'un espion fasse l'acquisition à la fois du message SOAP contenant une demande et du message SOAP contenant la réponse correspondante. Cette acquisition expose à la fois la nature de la demande et les détails de la réponse, qui peuvent inclure une ou plusieurs assertions.

L'exposition des détails de la demande sera dans certains cas un affaiblissement de la sécurité du demandeur en révélant des précisions sur le type d'assertions qu'il requiert, ou de qui ces assertions sont requises. Par exemple, si un espion peut déterminer que le site X demande fréquemment des assertions d'authentification avec une méthode de confirmation donnée du site Y, il peut être capable d'utiliser ces informations pour aider à compromettre le site X.

De même, l'espionnage sur une série d'interrogations d'autorisation peut permettre de créer une "carte" des ressources placées sous le contrôle d'une autorité d'autorisation donnée.

De plus, dans certains cas l'exposition de la demande elle-même pourrait constituer une violation de confidentialité. Par exemple, l'espionnage sur une interrogation et sa réponse peut exposer qu'un utilisateur donné est actif sur le site d'interrogation, ce qui pourrait être une information qui ne devrait pas être divulguée dans des cas tels que des sites d'informations médicales, des sites politiques, et ainsi de suite. Les détails de toute assertion portée dans la réponse peuvent être des information qui devraient être gardées confidentielles. Ceci est particulièrement vrai pour les réponses qui contiennent des assertions d'attribut; si ces attributs représentent des informations qui ne devraient pas être disponibles aux entités qui ne sont pas parties à la transaction (taux de crédit, attributs médicaux, et ainsi de suite), le risque d'espionnage est alors élevé.

Contre-mesures: dans les cas où l'un des risques pose problème, la contre-mesure des attaques d'espionnage est de fournir une forme de confidentialité du message en transit. Pour les messages SOAP, cette confidentialité peut être mise en application au niveau SOAP ou au niveau du transport SOAP (ou un niveau inférieur).

L'ajout de la confidentialité en transit au niveau SOAP signifie de construire le message SOAP de telle sorte que, indépendamment du transport SOAP, personne d'autre que le destinataire désigné ne soit capable d'accéder au message. La solution générale à ce problème est vraisemblablement le chiffrement XML. La présente Recommandation permet le chiffrement du message SOAP lui-même, ce qui élimine le risque d'espionnage, sauf si la clé utilisée pour le chiffrement a été compromise. Autrement, les développeurs peuvent s'appuyer sur la couche Transport de SOAP, ou une couche inférieure, pour fournir la confidentialité en transit.

Les détails de la façon de fournir cette confidentialité dépendent du transport spécifique SOAP choisi. Utiliser HTTP sur TLS/SSL est une méthode. D'autres transports nécessiteront d'autres techniques de confidentialité en transit; par exemple, un transport SMTP peut utiliser S/MIME.

Dans certains cas, une couche en-dessous du transport SOAP peut fournir la confidentialité en transit requise. Par exemple, si l'interaction demande-réponse est effectuée sur un tunnel IPSec, une confidentialité en transit adéquate peut être fournie par le tunnel lui-même.

2) Répétition

Menace: la vulnérabilité aux attaques de répétition est faible au niveau de la liaison SOAP. La répétition est plutôt un problème dans les différents profils. La question principale au sujet de la répétition au niveau de la liaison SOAP est le potentiel d'utilisation de la répétition comme méthode d'attaque de déni de service.

Contre-mesures: en général, le meilleur moyen pour empêcher les attaques de répétition est d'empêcher en premier lieu la capture du message. Certains des schémas de niveau transport utilisés pour fournir la confidentialité en transit vont atteindre ce but. Par exemple, si la conversation demande-réponse SAML survient avec SOAP sur HTTP/TLS, les tiers sont empêchés de capturer les messages.

Dans la mesure ou le répéteur potentiel n'a pas besoin de comprendre le message à répéter, des schémas tels que le chiffrement XML ne fournissent pas de protection contre la répétition. Si un attaquant peut capturer une demande SAML qui a été signée par le demandeur et chiffrée chez le répondant, l'attaquant peut alors répéter cette demande à tout moment sans qu'il soit nécessaire qu'il soit capable de la déchiffrer. La demande SAML inclut des informations sur l'heure de production de la demande, permettant de déterminer si une répétition survient. Autrement, la clé unique de la demande (son `ID`) peut être utilisée pour déterminer si elle est une demande répétée ou non.

Des menaces supplémentaires de la part des attaques en répétition incluent les cas où est en place un modèle "payé à la pièce". La répétition peut être utilisée pour faire payer de grosses charges sur un compte donné.

De même, dans les modèles où un nombre fixe d'interactions avec un système est alloué à (ou sont achetés par) un client, l'attaque en répétition pourrait les épuiser si le producteur ne veille pas à garder la trace de la clé unique de chaque demande.

3) Insertion de message

Menace: une demande ou réponse fabriquée est insérée dans le flux de messages. Une fausse réponse telle qu'un "oui" parasite en réponse à une interrogation de décision d'autorisation, ou retourner de fausses informations d'attribut en réponse à une interrogation d'attribut peut résulter en une action inappropriée du receveur.

Contre-mesures: la capacité à insérer une demande n'est pas une menace au niveau de la liaison SOAP. La menace d'insérer une fausse réponse peut être une attaque de déni de service, par exemple en retournant des fautes SOAP en réponses, mais cette attaque deviendrait vite évidente. L'attaque plus subtile qui consiste à retourner des réponses fabriquées est visée dans le protocole SAML; elle est appropriée car conformément à la définition de la liaison SOAP, chaque réponse SOAP doit contenir une seule réponse de protocole SAML sauf si elle contient une faute. Le protocole SAML traite cela avec deux mécanismes, la corrélation des réponses aux demandes en utilisant l'attribut obligatoire `InResponseTo`, ce qui rend une attaque plus difficile car une demande doit être interceptée pour générer des réponses, et par la prise en charge de l'authentification d'origine, via des réponses SAML signées ou par une connexion de transport sécurisée telle que SSL/TLS.

4) Suppression de message

Menace: l'attaque de suppression de message peut empêcher une demande d'atteindre un destinataire, ou empêcher la réponse d'atteindre le demandeur.

Contre-mesures: dans les deux cas, la liaison SOAP ne traite pas cette menace. En général, la corrélation des messages de demande et de réponse peut dissuader de telles attaques, par exemple par l'utilisation de l'attribut InResponseTo dans le StatusResponseType.

5) Modification de message

Menace: la modification de message est une menace pour la liaison SOAP dans les deux directions.

La modification de la demande pour en altérer les détails peut résulter en différences significatives dans la réponse retournée, qui peut à son tour être utilisée par un attaquant habile pour compromettre les systèmes qui dépendent des assertions retournées. Par exemple, altérer la liste des attributs requis dans les éléments `<Attribute>` peut produire des résultats conduisant à compromettre ou rejeter la demande par le répondant.

La modification de la demande pour altérer le producteur apparent de la demande pourrait résulter en déni de service ou acheminement incorrect de la réponse. Cette altération devrait survenir au-dessous du niveau SAML et est donc en dehors de son domaine d'application.

La modification de la réponse pour altérer les détails des assertions qu'elle contient pourrait avoir pour résultat de forts risques de compromission. Les simples exemples d'altération des détails d'une authentification ou d'une décision d'autorisation pourraient conduire à de très sérieuses atteintes à la sécurité.

Contre-mesures: afin de traiter ces menaces potentielles, on doit utiliser un système qui garantisse l'intégrité du message en transit. Le protocole SAML et la liaison SOAP n'exigent ni n'interdisent ni l'un ni l'autre le développement de systèmes qui garantissent l'intégrité de message en transit, mais à cause de cette vaste menace, il est fortement recommandé qu'un tel système soit utilisé. Au niveau de la liaison SOAP, cela peut être accompli par la signature numérique des demandes et réponses avec un système comme XML Signature.

Si les messages sont signés numériquement, le receveur a alors la garantie que le message n'a pas été altéré dans le transit, sauf si la clé utilisée a été compromise.

Le but de l'intégrité de message en transit peut aussi être réalisé à un niveau inférieur en utilisant un transport SOAP qui fournit la propriété de garantir l'intégrité, ou est fondé sur un protocole qui fournit une telle propriété. SOAP sur HTTP avec TLS/SSL est un transport qui fournirait une telle garantie.

Le chiffrement seul ne fournit pas cette protection, car même si le message intercepté ne peut pas être altéré *en soi*, il pourrait être remplacé par un message nouvellement créé.

6) Intrusion

Menace: la liaison SOAP est susceptible d'attaques par intrusion (MITM, *man-in-the-middle*). Pour empêcher des entités malveillantes d'opérer des intrusions (avec tous les périls discutés dans les deux paragraphes sur l'espionnage et la modification du message), une forme d'authentification bilatérale est nécessaire.

Contre-mesures: un système d'authentification bilatérale permettrait aux deux parties de déterminer que ce qu'elles voient dans une conversation vient réellement de l'autre partie à la conversation.

Au niveau de la liaison SOAP, cet objectif peut aussi être atteint par la signature numérique des demandes et des réponses. Cette méthode n'empêche pas un espion de s'installer au milieu et de retransmettre des deux côtés, mais il sera détecté s'il tente d'altérer la conversation.

Comme de nombreuses applications de SOAP n'utilisent pas de sessions, cette sorte d'authentification d'auteur (par opposition à l'authentification d'expéditeur) peut devoir être combinée avec des informations provenant de la couche transport pour confirmer que l'envoyeur et l'auteur sont la même partie afin d'empêcher une forme plus faible de "MITM comme espion".

Une autre implémentation dépendrait d'un transport SOAP fournissant l'authentification bilatérale, ou serait développée sur une couche inférieure qui la fournisse. Cela est encore illustré par SOAP sur HTTP avec TLS/SSL avec exigence des deux certificats côté serveur et côté client.

De plus, l'intervalle de validité des assertions retournées fonctionne comme un ajustement sur le niveau de risque d'attaques MITM. Plus courte est la fenêtre de validité de l'assertion, moindres seront les dommages qui peuvent être causés par une interception.

7) Utilisation de SOAP sur HTTP

Dans la mesure où la liaison SOAP exige des applications conformes qu'elles prennent en charge HTTP sur TLS/SSL avec un certain nombre de méthodes d'authentification bilatérale différentes telles que Basic sur SSL côté serveur et l'authentification appuyée sur des certificats sur SSL côté serveur, ces méthodes sont toujours disponibles pour atténuer les menaces dans les cas où d'autres systèmes de niveau inférieur ne sont pas disponibles et où les attaques énumérées ci-dessus sont considérées comme des menaces significatives.

Cela ne signifie pas que l'utilisation de HTTP sur TLS avec des formes d'authentification bilatérale soit obligatoire. Si on peut atteindre un niveau de protection acceptable contre les divers risques par d'autres moyens (par exemple, par un tunnel IPSec), le plein TLS avec certificats n'est pas exigé. Cependant, dans la majorité des cas pour SOAP sur HTTP, l'utilisation de HTTP sur TLS avec authentification bilatérale sera le choix approprié.

La RFC sur l'authentification HTTP (RFC 2617 de l'IETF) décrit les attaques possibles dans l'environnement HTTP lorsque les schémas d'authentification de base ou de résumé de message sont utilisés.

Cependant, l'utilisation de la sécurité au niveau transport (comme les protocoles SSL ou TLS sous HTTP) ne fournit la confidentialité et/ou l'intégrité et/ou l'authentification que pour "un saut". Pour les modèles où il peut y avoir des intermédiaires, ou où les assertions en question ont besoin de vivre sur plus d'un saut, l'utilisation de HTTP avec TLS/SSL ne procure pas la sécurité adéquate.

I.7.2 Profils de navigateur de la toile à signature unique (SSO, *single sign on*)

L'authentification d'utilisateur au site de source est explicitement hors domaine, comme le sont les questions se rapportant à l'authentification de ce site de source. La notion clé est que l'entité système de source doit être capable de certifier que l'entité système du client authentifié avec laquelle il interagit est la même que celle de l'étape de la

prochaine interaction. Une façon de réaliser cela est que ces étapes initiales soient effectuées en utilisant TLS comme couche de session sous-jacente au protocole utilisé pour cette interaction initiale (vraisemblablement HTTP).

I.7.2.1 Profil SSO

1) Espionnage

Menace: la possibilité d'espionnage existe dans tous les cas de navigateur de la toile.

Contre-mesures: dans les cas où la confidentialité est nécessaire (en gardant présent à l'esprit que toute assertion qui n'est pas envoyée de façon sécurisée, ainsi que les demandes qui lui sont associées, est disponible pour un espion malveillant), le trafic HTTP doit avoir lieu sur un transport qui assure la confidentialité. HTTP sur TLS/SSL et le protocole de sécutité IP satisfont à cette exigence.

2) Vol des informations d'authentification de l'usager

Menace: dans le cas où le sujet s'authentifie auprès du site de source en révélant des informations d'authentification réutilisables, par exemple, sous la forme d'un mot de passe, le vol des informations d'authentification permettra à un adversaire de se faire passer pour le sujet.

Contre-mesures: pour éviter ce problème, la connexion entre le navigateur du sujet et le site de source doit implémenter une sauvegarde de confidentialité. De plus, des mesures doivent être prises par le sujet ou par le site de destination pour s'assurer que le site de source est vraiment celui attendu et un site source de confiance avant de révéler les informations d'authentification. HTTP sur TLS peut être utilisé pour traiter cette question.

3) Vol du jeton du titulaire

Menace: dans le cas où l'assertion d'authentification contient l'identifiant de protocole d'authentification du titulaire de l'assertion, le vol de l'artifice permettra à un adversaire de se faire passer pour le sujet.

Contre-mesures: chacune des méthodes suivantes diminue la probabilité que cela arrive:

Le site de destination implémente une sauvegarde de confidentialité sur sa connexion avec le navigateur du sujet.

Le sujet ou le site de destination s'assure (hors bande) que le site de source implémente une sauvegarde de confidentialité sur sa connexion avec le navigateur du sujet.

Le site de destination vérifie que le navigateur du sujet a été directement redirigé par un site de source qui a directement authentifié le sujet.

Le site de source refuse de répondre à plus d'une demande pour une assertion correspondant au même identifiant d'assertion.

Si l'assertion contient un élément de condition du type **AudienceRestrictionType** qui identifie un domaine spécifique, le site de destination vérifie alors qu'il est membre de ce domaine.

La connexion entre le site de destination et le site de source, sur laquelle l'identifiant d'assertion est passé, est implémentée avec une sauvegarde de confidentialité.

Le site de destination, dans sa communication avec le site de source, sur laquelle l'identifiant d'assertion est passé, doit vérifier que le site de source est véritablement le site de source attendu et de confiance.

4) Répétition

La possibilité d'une attaque en répétition existe pour cet ensemble de profils. Une attaque en répétition peut être utilisée pour tenter un déni de service ou pour récupérer frauduleusement des informations. Les contre-mesures spécifiques dépendent de la liaison spécifique qui est utilisée et sont exposées ci-dessus.

5) Insertion de message

L'attaque d'insertion de message est exposée au § I.7.1.

6) Suppression de message

Menace: la suppression d'un message durant toute étape de l'interaction entre le navigateur, le producteur d'assertion SAML, et le consommateur d'assertions SAML causera l'échec de l'interaction. Il en résulte un déni de certains services mais cela n'accroît l'exposition d'aucune information au risque.

Contre-mesures: l'utilisation d'un canal de transport protégé en intégrité traite la menace de suppression de message lorsqu'aucun intermédiaire n'est présent.

7) Modification du message

Menace: la possibilité d'altération des messages dans le flux existe pour cet ensemble de profils. Certains résultats indésirables potentiels sont:

l'altération de la demande initiale peut avoir pour résultat le rejet chez le producteur SAML, ou la création d'un artifice visant une ressource différente de celle demandée.

L'altération de l'artifice peut avoir pour résultat un déni de service chez le consommateur SAML.

L'altération des assertions elles mêmes pendant le transit pourrait avoir pour résultat toutes sortes de mauvais résultats (si elles ne sont pas signées) ou déni de service (si elles sont signées et que le consommateur les rejette).

Contre-mesures: pour éviter les modifications de message, le trafic doit être transporté au moyen d'un système qui garantisse l'intégrité du message de point d'extrémité à point d'extrémité.

Pour les profils fondés sur un navigateur de la toile, la méthode recommandée de fourniture de l'intégrité du message en transit est l'utilisation de HTTP sur TLS/SSL avec une suite de chiffrement qui fournisse la vérification d'intégrité des données.

8) Intrusion

Menace: les attaques par intrusion sont particulièrement pernicieuses pour cet ensemble de profils. Le MITM peut relayer la demande, capturer l'assertion (ou l'artifice) retournée, et en relayer une fausse en retour. L'utilisateur d'origine ne peut alors pas accéder à la ressource en question, mais le MITM peut le faire en utilisant la ressource capturée.

Contre-mesures: empêcher cette menace exige un certain nombre de contre-mesures. D'abord, d'utiliser un système qui fournisse une forte authentification bilatérale rendra beaucoup plus difficile au MITM de s'insérer dans la conversation.

Cependant il existe toujours la possibilité qu'un MITM agisse uniquement comme transmetteur d'accès bidirectionnel, et espionne les informations dans l'intention de capturer l'assertion retournée ou de la manipuler (et éventuellement d'altérer le retour final au demandeur). Mettre en place un système de confidentialité empêchera l'espionnage. Mettre en place un système d'intégrité des données empêchera l'altération du message durant la transmission d'accès.

Pour cet ensemble de profils, toutes les exigences de forte authentification bilatérale de session, de confidentialité, et d'intégrité des données doivent être satisfaites par l'utilisation de HTTP sur TLS/SSL si la couche TLS/SSL utilise une suite de chiffrement appropriée (assez forte pour fournir la confidentialité, et prenant en charge l'intégrité des données) et en exigeant des certificats X.509 v3 pour l'authentification.

9) Fausse identité sans ré-authentification

Menace: un escroc essaye de se faire passer pour un principal légitimement en cours de session et obtenir ainsi l'accès à des ressources protégées.

Une fois qu'un principal a réussi à ouvrir une session chez un fournisseur d'identité, les messages `<AuthnRequest>` suivants provenant de différents fournisseurs de service concernant ce Principal ne causeront pas nécessairement la ré-authentification du principal. Les principaux doivent, cependant, être authentifiés sauf si le fournisseur d'identité peut déterminer qu'un `<AuthnRequest>` est associé non seulement à l'identité du principal, mais aussi à une session de fournisseur d'identité validement authentifiée pour ce principal.

Contre-mesures: dans les implémentations où cette menace pose problème, les fournisseurs d'identité doivent entretenir des informations d'état concernant les sessions actives, et doivent valider la correspondance entre un `<AuthnRequest>` et une session active avant de produire une `<Response>` sans d'abord authentifier le principal. Les mouchard postés par les fournisseurs d'identité peuvent être utilisés pour prendre en charge ce processus de validation, bien que la Liberté n'oblige pas à une approche fondée sur le mouchard.

I.7.2.2 Profil de client et mandataire amélioré

1) Intrusion

Menace: interception des messages SOAP `AuthnRequest` et `Response`, permettant ultérieurement de se faire passer pour le principal.

Une entité système parasite peut s'instituer elle-même comme intrus (MITM) entre le client amélioré et un fournisseur de service légitime, et agir dans le rôle du fournisseur de service dans les interactions avec le client amélioré, et dans le rôle du client amélioré dans les interactions avec le fournisseur de service légitime. De cette façon, dans une première étape, le MITM est capable d'intercepter le `AuthnRequest` du fournisseur de service et de substituer tout URL de son choix pour la valeur de `responseConsumerServiceURL` dans le bloc d'en-tête PAOS avant de transmettre le `AuthnRequest` au client amélioré. Normalement le MITM insèrera une valeur d'URL qui renvoie sur lui-même. Ensuite, si le client amélioré reçoit une réponse de la part du fournisseur d'identité et ensuite envoie la réponse

contenue au `responseConsumerServiceURL` reçu du MITM, celui-ci sera capable de se déguiser en principal chez le fournisseur de service légitime.

Contre-mesure: le fournisseur d'identité spécifie au client amélioré l'adresse à laquelle le client amélioré doit envoyer la réponse. Le `responseConsumerServiceURL` dans l'en-tête PAOS n'est utilisé que pour les réponses d'erreur de la part du client amélioré – comme spécifié dans le profil.

2) Déni de service

Menace: changer une demande SOAP `AuthnRequest` de telle sorte qu'elle ne puisse plus être traitée, comme en changeant la valeur de l'attribut de service du bloc d'en-tête PAOS en une valeur inconnue ou en changeant le bloc d'en-tête ECP `ProviderID` ou `IDPList` pour causer l'échec de la demande.

Contre-mesures: fournir la protection de l'intégrité du message SOAP en utilisant la sécurité de message SOAP ou SSL/TLS.

I.7.2.3 Profil de découverte de fournisseur d'identité

Menace: attaque d'empoisonnement de mouchard, dans laquelle les paramètre intérieurs du mouchard sont modifiés, pour provoquer, par exemple, la découverte d'un fournisseur d'identité frauduleux.

Contre-mesures: le mécanisme spécifique d'utilisation d'un domaine commun limite la faisabilité de cette menace.

I.7.2.4 Profil unique de terminaison de session

Menace: un attaquant passif peut collecter l'identifiant de nom d'un principal.

Durant les étapes initiales, un attaquant passif peut collecter les informations de `<LogoutRequest>` lorsqu'il est produit dans le renvoi. Exposer ces données fait peser une menace sur la confidentialité.

Contre-mesures: tous les échanges devraient être effectués sur un transport sécurisé tel que SSL ou TLS.

Menace: message `<LogoutRequest>` non signé

Un `<LogoutRequest>` non signé pourrait être injecté par une entité système parasite déniant ainsi le service au Principal. En supposant que le `NameID` peut être déduit ou dérivé, il est alors concevable que l'agent d'utilisateur soit amené à délivrer un message `<LogoutRequest>` fabriqué.

Contre-mesures: signer le message <LogoutRequest>. Le fournisseur d'identité peut aussi vérifier l'identité d'un Principal en l'absence d'une demande signée.

I.7.2.5 Profils de gestion d'identifiant de nom

Menace: permettre aux entités système de corréler les informations ou autrement d'exposer de façon inappropriée les informations d'identité, causant par là une atteinte à la confidentialité.

Contre-mesures: l'IDP doit veiller à utiliser des identifiants de nom différents avec différents fournisseurs de service pour le même principal. L'IDP devrait chiffrer l'identifiant de nom qu'il retourne au fournisseur de service, permettant aux interactions ultérieures d'utiliser un identifiant opaque.

I.7.2.6 Profils d'attribut

Les menaces qui se rapportent aux liaisons associées aux profils d'attribut sont exposées plus haut. Aucune menace spécifique du profil spécifique n'est connue.

Appendice II

Enregistrement du type de support MIME application/samlassertion+xml

Le présent appendice contient l'enregistrement du type de support MIME Application/Assertion de SAML.

Nom de type de support MIME

– `application`

Nom de sous-type MIME

– `samlassertion+xml`

Paramètres exigés

– Aucun

Paramètres facultatifs

– `charset`

– Identique au paramètre `charset` de `application/xml` dans la RFC 3923 de l'IETF.

Considérations sur le codage

– Comme pour application/xml dans la RFC 3923 de l'IETF.

Considérations sur la sécurité

Les objets `samlassertion+xml-typed` ne contiennent aucun contenu exécutable. Cependant, les assertions SAML sont des objets fondés sur XML. Comme tels, ils relèvent de toutes les considérations générales sur la sécurité présentées dans le § 10 de la RFC 3923 de l'IETF, ainsi que les considérations supplémentaires, car ce sont des objets de sécurité explicites. Par exemple, les objets `samlassertion+xml-`typed contiendront souvent des données qui peuvent identifier ou appartenir à des personnes physiques, et peuvent être utilisées comme bases de décisions de sessions et de contrôle d'accès.

Pour contrer des problèmes potentiels, les objets `samlassertion+xml-`typed contiennent des données qui devraient être signées de façon appropriée par l'envoyeur. Toute signature de cette sorte doit être vérifiée par le receveur des données – à la fois comme signature valide et comme étant la signature de l'envoyeur. Les producteurs d'objets `samlassertion+xml-`typed contenant les assertions SAML peuvent aussi chiffrer toutes les assertions ou des portions des assertions.

De plus, les profils SAML et les liaisons de protocoles spécifient l'utilisation de canaux sécurisés en tant que de besoin.

SAML Version 2 (la présente Recommandation) incorpore dans sa conception diverses techniques de protection de la confidentialité. Par exemple: des opérateurs opaques, spécifiques des interactions entre entités système spécifiques, peuvent être assignés aux sujets. Les opérateurs sont transposables à des identifiants de contexte plus larges (par exemple, des adresses de messagerie électronique, des identifiants de comptes, etc.) par les seules parties spécifiques.

Considérations d'interopérabilité

Les assertions SAML ont un numéro de version explicite. Les consommateurs d'assertions devrait s'assurer qu'ils observent les informations de version d'assertion et se comportent en conséquence.

Spécification publiée

SAML Version 2 (la présente Recommandation) spécifie explicitement l'utilisation du type de support MIME `application/samlassertion+xml`. Cependant, il est concevable que les assertions non-SAML (c'est-à-dire, SAMLv1 et/ou SAMLv1.1) puissent en pratique être convoyées en utilisant les liaisons SAML.

Applications qui utilisent ce type de support

Potentiellement toute application mettant en œuvre SAML, aussi bien que les applications qui implémentent des spécifications fondées sur SAML.

Informations supplémentaires

Nombre(s) magique(s)

En général, comme pour application/xml. En particulier, l'élément racine XML de l'objet retourné aura un nom qualifié d'espace de nom avec:

- un nom local de: `Assertion`;
- un URI d'espace de nom de: un des URI d'espace de nom XML d'assertion SAML spécifique de la version, comme défini par la Recommandation appropriée "centrale" SAML spécifique de la version.

Spécifiquement avec SAML, l'élément racine de l'objet retourné peut être `<saml:Assertion>` ou `<saml:EncryptedAssertion>`, où "saml" représente tout préfixe d'espace de nom XML qui se transpose en URI d'espace de nom d'assertion SAML:

urn:oasis:names:tc:SAML:2.0:assertion

Extension(s) de fichier

Aucune.

Codes de type de fichier Macintosh

Aucun.

Adresse personnelle & de messagerie électronique de la personne à contacter pour des informations complémentaires

Cet enregistrement a été fait au nom de OASIS Security Services Technical Committee (SSTC).

Usage de destination

Commun.

Appendice III

Enregistrement du type de support MIME application/samlmetadata+xml

Le présent appendice définit un type de support MIME `application/samlmetadata+xml` à utiliser avec la mise en série XML des métadonnées du Langage de balisage d'assertions de sécurité.

1) Nom de type de support MIME

– `application`

2) Nom de sous-type MIME

– `Samlmetadata+xml`

3) Paramètres exigés

– Aucun

4) Paramètres facultatifs

– `charset`

– Le même que le paramètre `charset` de application/xml (voir la RFC 3023 de l'IETF).

5) Considérations sur le codage

– Les mêmes que pour `application/xml` dans la RFC 3023 de l'IETF.

6) Considérations sur la sécurité

Les objets du type `samlmetadata+xml` ne contiennent pas de contenu exécutable. Cependant, ces objets sont fondés sur XML, et donc, ils sont soumis à toutes les considérations générales sur la sécurité présentées dans le § 10 de la RFC 3023 de l'IETF.

Pour contrer les problèmes potentiels, l'éditeur peut signer les objets du type `samlmetadata+xml`. Toute signature de cette sorte devrait être vérifiée par le receveur de données – à la fois comme signature valide et comme étant la signature de l'éditeur.

7) Considérations sur l'interopérabilité

Les métadonnées SAML prennent explicitement en charge l'identification des protocoles et versions acceptées par les entités identifiées. Par exemple, une entité de fournisseur d'identité peut être répertoriée comme prenant en charge SAML v2.0 et d'autres protocoles si ils sont identifiables sans ambiguïté via l'URI. Ce protocole prend en charge les informations convoyées via l'attribut `protocolSupportEnumeration` des objets de métadonnées de **RoleDescriptorType**.

8) Recommandation publiée

Les métadonnées SAML spécifient explicitement l'utilisation du type de support `application/samlmetadata+xml` MIME.

Applications qui utilisent ce type de support:

potentiellement toutes les applications qui implémentent SAML v2.0, aussi bien que les applications qui implémentent les spécifications fondées sur SAML.

9) Informations supplémentaires

1) Nombres magiques

En général, les mêmes que pour `application/xml` dans la RFC 3023 de l'IETF. En particulier, l'élément racine XML de l'objet retourné aura un nom qualifié d'espace de nom avec:

– un nom local de: EntityDescriptor, ou AffiliationDescriptor, ou EntitiesDescriptor;

– un URI d'espace de nom de: urn:oasis:names:tc:SAML:2.0:metadata (l'espace de nom de métadonnées de SAMLv2.0).

10) Extensions de fichier

Aucune.

11) Codes de type de fichier Macintosh

Aucune.

12) Adresse personnelle & de messagerie électronique de la personne à contacter pour des informations complémentaires

Cet enregistrement a été fait au nom de OASIS Security Services Technical Committee (SSTC).

13) Usage de destination

Commun.

Appendice IV

Utilisation de SSL

Certaines implémentations de SAML peuvent accepter l'utilisation de SSL 3.0 en plus ou en remplacement de TLS 1.0. Les implémentations qui utilisent SSL 3.0 devraient s'assurer que la sécurité globale de l'implémentation est cohérente avec les restrictions imposées à l'utilisation du chiffrement dans TLS. Par exemple, l'exigence d'utiliser la suite de chiffrement TLS_RSA_WITH_3DES_EDE_CBC_SHA traduite en utilisation de la suite de chiffrement SSL_RSA_WITH_3DES_EDE_CBC_SHA. Les mises en œuvre FIPS à capacité SSL utilisent la suite de chiffrement FIPS correspondant à la suite de chiffrement SSL SSL_RSA_WITH_3DES_EDE_CBC_SHA.

Les implémentations TLS du profil SSO de la toile de SAML qui acceptent la suite de chiffrement TLS_RSA_WITH_3DES_EDE_CBC_SHA utiliseront la suite de chiffrement SSL_RSA_WITH_3DES_EDE_CBC_SHA.

Appendice V

Schéma SAML de contexte d'authentification

Le présent appendice contient le schéma SAML de contexte d'authentification pour le certificat SSL (sslcert).

```
<?xml version="1.0" encoding="UTF-8"?>

<xs:schema targetNamespace="urn:oasis:names:tc:SAML:2.0:ac:classes:TLSClient"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns="urn:oasis:names:tc:SAML:2.0:ac:classes:TLSClient"
  finalDefault="extension"
  blockDefault="substitution"
  version="2.0">

  <xs:redefine schemaLocation="saml-schema-authn-context-types-2.0.xsd">

    <xs:annotation>
      <xs:documentation>
        Class identifier: urn:oasis:names:tc:SAML:2.0:ac:classes:TLSClient
        Document identifier: saml-schema-authn-context-sslcert-2.0
        Location: http://docs.oasis-open.org/security/saml/v2.0/
        Revision history:
          V2.0 (March, 2005):
            New authentication_u99 context class schema for SAML V2.0.
      </xs:documentation>
    </xs:annotation>

    <xs:complexType name="AuthnContextDeclarationBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthnContextDeclarationBaseType">
          <xs:sequence>
            <xs:element ref="Identification" minOccurs="0"/>
            <xs:element ref="TechnicalProtection" minOccurs="0"/>
            <xs:element ref="OperationalProtection" minOccurs="0"/>
            <xs:element ref="AuthnMethod"/>
            <xs:element ref="GoverningAgreements" minOccurs="0"/>
            <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
          </xs:sequence>
          <xs:attribute name="ID" type="xs:ID" use="optional"/>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>
```

```
    <xs:complexType name="AuthnMethodBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthnMethodBaseType">
          <xs:sequence>
            <xs:element ref="PrincipalAuthenticationMechanism"/>
            <xs:element ref="Authenticator"/>
            <xs:element ref="AuthenticatorTransportProtocol" minOccurs="0"/>
            <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="PrincipalAuthenticationMechanismType">
      <xs:complexContent>
        <xs:restriction base="PrincipalAuthenticationMechanismType">
          <xs:sequence>
            <xs:element ref="RestrictedPassword"/>
          </xs:sequence>
          <xs:attribute name="preauth" type="xs:integer" use="optional"/>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="AuthenticatorBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthenticatorBaseType">
          <xs:sequence>
            <xs:element ref="DigSig"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="PublicKeyType">
      <xs:complexContent>
        <xs:restriction base="PublicKeyType">
          <xs:attribute name="keyValidation" type="xs:anyURI"
fixed="urn:oasis:names:tc:SAML:2.0:ac:classes:X509"/>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="AuthenticatorTransportProtocolType">
      <xs:complexContent>
        <xs:restriction base="AuthenticatorTransportProtocolType">
          <xs:sequence>
            <xs:choice>
              <xs:element ref="SSL"/>
              <xs:element ref="WTLS"/>
            </xs:choice>
            <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

  </xs:redefine>

</xs:schema>
```

Appendice VI

Schéma XML des types de contexte d'authentification

Le présent appendice fait la liste complète du schéma XML de types de contexte d'authentification et du schéma XML de contexte d'authentification lui-même, utilisé pour la validation des déclarations généralisées individuelles. Le schéma types n'a pas d'espace de nom cible lui-même, et est donc inclus dans l'Appendice V.

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  elementFormDefault="qualified"
  version="2.0">

  <xs:annotation>
    <xs:documentation>
      Document identifier: saml-schema-authn-context-types-2.0
      Location: http://docs.oasis-open.org/security/saml/v2.0/
      Revision history:
          V2.0 (March, 2005):
          New core authentication context schema types for SAML V2.0.
    </xs:documentation>
  </xs:annotation>

  <xs:element name="AuthenticationContextDeclaration"
type="AuthnContextDeclarationBaseType">
    <xs:annotation>
      <xs:documentation>
        A particular assertion on an identity
        provider's part with respect to the authentication
        context associated with an authentication assertion.
      </xs:documentation>
    </xs:annotation>
  </xs:element>

  <xs:element name="Identification" type="IdentificationType">
    <xs:annotation>
      <xs:documentation>
        Refers to those characteristics that describe the
        processes and mechanisms
        the Authentication Authority uses to initially create
        an association between a Principal
        and the identity (or name) by which the Principal will
        be known
      </xs:documentation>
    </xs:annotation>
  </xs:element>

  <xs:element name="PhysicalVerification">
    <xs:annotation>
      <xs:documentation>
        This element indicates that identification has been
        performed in a physical
        face-to-face meeting with the principal and not in an
        online manner.
      </xs:documentation>
    </xs:annotation>
    <xs:complexType>
      <xs:attribute name="credentialLevel">
        <xs:simpleType>
          <xs:restriction base="xs:NMTOKEN">
            <xs:enumeration value="primary"/>
            <xs:enumeration value="secondary"/>
          </xs:restriction>
        </xs:simpleType>
      </xs:attribute>
    </xs:complexType>
  </xs:element>
```

```
<xs:element name="WrittenConsent" type="ExtensionOnlyType"/>

<xs:element name="TechnicalProtection" type="TechnicalProtectionBaseType">
  <xs:annotation>
    <xs:documentation>
      Refers to those characterstics that describe how the
      'secret' (the knowledge or possession
      of which allows the Principal to authenticate to the
      Authentication Authority) is kept secure
    </xs:documentation>
  </xs:annotation>
</xs:element>

<xs:element name="SecretKeyProtection" type="SecretKeyProtectionType">
  <xs:annotation>
    <xs:documentation>
      This element indicates the types and strengths of
      facilities
      of a UA used to protect a shared secret key from
      unauthorized access and/or use.
    </xs:documentation>
  </xs:annotation>
</xs:element>

<xs:element name="PrivateKeyProtection" type="PrivateKeyProtectionType">
  <xs:annotation>
    <xs:documentation>
      This element indicates the types and strengths of
      facilities
      of a UA used to protect a private key from
      unauthorized access and/or use.
    </xs:documentation>
  </xs:annotation>
</xs:element>

<xs:element name="KeyActivation" type="KeyActivationType">
  <xs:annotation>
    <xs:documentation>The actions that must be performed
      before the private key can be used. </xs:documentation>
  </xs:annotation>
</xs:element>

<xs:element name="KeySharing" type="KeySharingType">
  <xs:annotation>
    <xs:documentation>Whether or not the private key is shared
      with the certificate authority.</xs:documentation>
  </xs:annotation>
</xs:element>

<xs:element name="KeyStorage" type="KeyStorageType">
  <xs:annotation>
    <xs:documentation>
      In which medium is the key stored.
      memory - the key is stored in memory.
      smartcard - the key is stored in a smartcard.
      token - the key is stored in a hardware token.
      MobileDevice - the key is stored in a mobile device.
      MobileAuthCard - the key is stored in a mobile
      authentication card.
    </xs:documentation>
  </xs:annotation>
</xs:element>

<xs:element name="SubscriberLineNumber" type="ExtensionOnlyType"/>
<xs:element name="UserSuffix" type="ExtensionOnlyType"/>

<xs:element name="Password" type="PasswordType">
  <xs:annotation>
    <xs:documentation>
      This element indicates that a password (or passphrase)
```

```
      has been used to
      authenticate the Principal to a remote system.
    </xs:documentation>
  </xs:annotation>
</xs:element>

<xs:element name="ActivationPin" type="ActivationPinType">
  <xs:annotation>
    <xs:documentation>
      This element indicates that a Pin (Personal
      Identification Number) has been used to authenticate the Principal
      to some local system in order to activate a key.
    </xs:documentation>
  </xs:annotation>
</xs:element>

<xs:element name="Token" type="TokenType">
  <xs:annotation>
    <xs:documentation>
      This element indicates that a hardware or software
      token is used
      as a method of identifying the Principal.
    </xs:documentation>
  </xs:annotation>
</xs:element>

<xs:element name="TimeSyncToken" type="TimeSyncTokenType">
  <xs:annotation>
    <xs:documentation>
      This element indicates that a time synchronization
      token is used to identify the Principal. hardware -
      the time synchronization
      token has been implemented in hardware. software - the
      time synchronization
      token has been implemented in software. SeedLength -
      the length, in bits, of the
      random seed used in the time synchronization token.
    </xs:documentation>
  </xs:annotation>
</xs:element>

<xs:element name="Smartcard" type="ExtensionOnlyType">
  <xs:annotation>
    <xs:documentation>
      This element indicates that a smartcard is used to
      identify the Principal.
    </xs:documentation>
  </xs:annotation>
</xs:element>

<xs:element name="Length" type="LengthType">
  <xs:annotation>
    <xs:documentation>
      This element indicates the minimum and/or maximum
      ASCII length of the password which is enforced (by the UA or the
      IdP). In other words, this is the minimum and/or maximum number of
      ASCII characters required to represent a valid password.
      min - the minimum number of ASCII characters required
      in a valid password, as enforced by the UA or the IdP.
      max - the maximum number of ASCII characters required
      in a valid password, as enforced by the UA or the IdP.
    </xs:documentation>
  </xs:annotation>
</xs:element>

<xs:element name="ActivationLimit" type="ActivationLimitType">
  <xs:annotation>
    <xs:documentation>
      This element indicates the length of time for which an
      PIN-based authentication is valid.
    </xs:documentation>
```

```
    </xs:annotation>
  </xs:element>

  <xs:element name="Generation">
    <xs:annotation>
      <xs:documentation>
        Indicates whether the password was chosen by the
        Principal or auto-supplied by the Authentication Authority.
        principalchosen - the Principal is allowed to choose
        the value of the password. This is true even if
        the initial password is chosen at random by the UA or
        the IdP and the Principal is then free to change
        the password.
        automatic - the password is chosen by the UA or the
        IdP to be cryptographically strong in some sense,
        or to satisfy certain password rules, and that the
        Principal is not free to change it or to choose a new password.
      </xs:documentation>
    </xs:annotation>

    <xs:complexType>
      <xs:attribute name="mechanism" use="required">
        <xs:simpleType>
          <xs:restriction base="xs:NMTOKEN">
            <xs:enumeration value="principalchosen"/>
            <xs:enumeration value="automatic"/>
          </xs:restriction>
        </xs:simpleType>
      </xs:attribute>
    </xs:complexType>
  </xs:element>

  <xs:element name="AuthnMethod" type="AuthnMethodBaseType">
    <xs:annotation>
      <xs:documentation>
        Refers to those characteristics that define the
        mechanisms by which the Principal authenticates to the
        Authentication Authority.
      </xs:documentation>
    </xs:annotation>
  </xs:element>

  <xs:element name="PrincipalAuthenticationMechanism"
type="PrincipalAuthenticationMechanismType">
    <xs:annotation>
      <xs:documentation>
        The method that a Principal employs to perform
        authentication to local system components.
      </xs:documentation>
    </xs:annotation>
  </xs:element>

  <xs:element name="Authenticator" type="AuthenticatorBaseType">
    <xs:annotation>
      <xs:documentation>
        The method applied to validate a principal's
        authentication across a network
      </xs:documentation>
    </xs:annotation>
  </xs:element>

  <xs:element name="ComplexAuthenticator" type="ComplexAuthenticatorType">
    <xs:annotation>
      <xs:documentation>
        Supports Authenticators with nested combinations of
        additional complexity.
      </xs:documentation>
    </xs:annotation>
  </xs:element>

  <xs:element name="PreviousSession" type="ExtensionOnlyType">
```

```
    <xs:annotation>
      <xs:documentation>
        Indicates that the Principal has been strongly
        authenticated in a previous session during which the IdP has set a
        cookie in the UA. During the present session the Principal has only
        been authenticated by the UA returning the cookie to the IdP.
      </xs:documentation>
    </xs:annotation>
  </xs:element>

  <xs:element name="ResumeSession" type="ExtensionOnlyType">
    <xs:annotation>
      <xs:documentation>
        Rather like PreviousSession but using stronger
        security. A secret that was established in a previous session with
        the Authentication Authority has been cached by the local system
        and is now re-used (e.g. a Master Secret is used to derive new
        session keys in TLS, WTLS).
      </xs:documentation>
    </xs:annotation>
  </xs:element>

  <xs:element name="ZeroKnowledge" type="ExtensionOnlyType">
    <xs:annotation>
      <xs:documentation>
        This element indicates that the Principal has been
        authenticated by a zero knowledge technique as specified in ISO/IEC
        9798-5.
      </xs:documentation>
    </xs:annotation>
  </xs:element>

  <xs:element name="SharedSecretChallengeResponse"
type="SharedSecretChallengeResponseType"/>

  <xs:complexType name="SharedSecretChallengeResponseType">
    <xs:annotation>
      <xs:documentation>
        This element indicates that the Principal has been
        authenticated by a challenge-response protocol utilizing shared
        secret keys and symmetric cryptography.
      </xs:documentation>
    </xs:annotation>
    <xs:sequence>
      <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:attribute name="method" type="xs:anyURI" use="optional"/>
  </xs:complexType>

  <xs:element name="DigSig" type="PublicKeyType">
    <xs:annotation>
      <xs:documentation>
        This element indicates that the Principal has been
        authenticated by a mechanism which involves the Principal computing
        a digital signature over at least challenge data provided
        by the IdP.
      </xs:documentation>
    </xs:annotation>
  </xs:element>

  <xs:element name="AsymmetricDecryption" type="PublicKeyType">
    <xs:annotation>
      <xs:documentation>
        The local system has a private key but it is used
        in decryption mode, rather than signature mode. For example, the
        Authentication Authority generates a secret and encrypts it using
        the local system's public key: the local system then proves it has
        decrypted the secret.
      </xs:documentation>
    </xs:annotation>
  </xs:element>
```

```
  <xs:element name="AsymmetricKeyAgreement" type="PublicKeyType">
    <xs:annotation>
      <xs:documentation>
        The local system has a private key and uses it for
        shared secret key agreement with the Authentication Authority
        (e.g., via Diffie Helman).
      </xs:documentation>
    </xs:annotation>
  </xs:element>

  <xs:complexType name="PublicKeyType">
    <xs:sequence>
      <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:attribute name="keyValidation" use="optional"/>
  </xs:complexType>

  <xs:element name="IPAddress" type="ExtensionOnlyType">
    <xs:annotation>
      <xs:documentation>
        This element indicates that the Principal has been
        authenticated through connection from a particular IP address.
      </xs:documentation>
    </xs:annotation>
  </xs:element>

  <xs:element name="SharedSecretDynamicPlaintext" type="ExtensionOnlyType">
    <xs:annotation>
      <xs:documentation>
        The local system and Authentication Authority
        share a secret key. The local system uses this to encrypt a
        randomised string to pass to the Authentication Authority.
      </xs:documentation>
    </xs:annotation>
  </xs:element>

  <xs:element name="AuthenticatorTransportProtocol"
type="AuthenticatorTransportProtocolType">
    <xs:annotation>
      <xs:documentation>
        The protocol across which Authenticator information is
        transferred to an Authentication Authority verifier.
      </xs:documentation>
    </xs:annotation>
  </xs:element>

  <xs:element name="HTTP" type="ExtensionOnlyType">
    <xs:annotation>
      <xs:documentation>
        This element indicates that the Authenticator has been
        transmitted using bare HTTP utilizing no additional security
        protocols.
      </xs:documentation>
    </xs:annotation>
  </xs:element>

  <xs:element name="IPSec" type="ExtensionOnlyType">
    <xs:annotation>
      <xs:documentation>
        This element indicates that the Authenticator has been
        transmitted using a transport mechanism protected by an IPSEC session.
      </xs:documentation>
    </xs:annotation>
  </xs:element>

  <xs:element name="WTLS" type="ExtensionOnlyType">
    <xs:annotation>
      <xs:documentation>
        This element indicates that the Authenticator has been
        transmitted using a transport mechanism protected by a WTLS session.
```

```
    </xs:documentation>
  </xs:annotation>
</xs:element>

<xs:element name="MobileNetworkNoEncryption" type="ExtensionOnlyType">
  <xs:annotation>
    <xs:documentation>
      This element indicates that the Authenticator has been
      transmitted solely across a mobile network using no additional
      security mechanism.
    </xs:documentation>
  </xs:annotation>
</xs:element>

<xs:element name="MobileNetworkRadioEncryption" type="ExtensionOnlyType"/>
<xs:element name="MobileNetworkEndToEndEncryption" type="ExtensionOnlyType"/>

<xs:element name="SSL" type="ExtensionOnlyType">
  <xs:annotation>
    <xs:documentation>
      This element indicates that the Authenticator has been
      transmitted using a transport mechanism protected by an SSL or TLS
      session.
    </xs:documentation>
  </xs:annotation>
</xs:element>

<xs:element name="PSTN" type="ExtensionOnlyType"/>
<xs:element name="ISDN" type="ExtensionOnlyType"/>
<xs:element name="ADSL" type="ExtensionOnlyType"/>

<xs:element name="OperationalProtection" type="OperationalProtectionType">
  <xs:annotation>
    <xs:documentation>
      Refers to those characteristics that describe
      procedural security controls employed by the Authentication Authority.
    </xs:documentation>
  </xs:annotation>
</xs:element>

<xs:element name="SecurityAudit" type="SecurityAuditType"/>
<xs:element name="SwitchAudit" type="ExtensionOnlyType"/>
<xs:element name="DeactivationCallCenter" type="ExtensionOnlyType"/>

<xs:element name="GoverningAgreements" type="GoverningAgreementsType">
  <xs:annotation>
    <xs:documentation>
      Provides a mechanism for linking to external (likely
      human readable) documents in which additional business agreements,
      (e.g., liability constraints, obligations, etc.) can be placed.
    </xs:documentation>
  </xs:annotation>
</xs:element>

<xs:element name="GoverningAgreementRef" type="GoverningAgreementRefType"/>

<xs:simpleType name="nymType">
  <xs:restriction base="xs:NMTOKEN">
    <xs:enumeration value="anonymity"/>
    <xs:enumeration value="verinymity"/>
    <xs:enumeration value="pseudonymity"/>
  </xs:restriction>
</xs:simpleType>

<xs:complexType name="AuthnContextDeclarationBaseType">
  <xs:sequence>
    <xs:element ref="Identification" minOccurs="0"/>
    <xs:element ref="TechnicalProtection" minOccurs="0"/>
    <xs:element ref="OperationalProtection" minOccurs="0"/>
    <xs:element ref="AuthnMethod" minOccurs="0"/>
    <xs:element ref="GoverningAgreements" minOccurs="0"/>
```

```
    <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
  </xs:sequence>
  <xs:attribute name="ID" type="xs:ID" use="optional"/>
</xs:complexType>

<xs:complexType name="IdentificationType">
  <xs:sequence>
    <xs:element ref="PhysicalVerification" minOccurs="0"/>
    <xs:element ref="WrittenConsent" minOccurs="0"/>
    <xs:element ref="GoverningAgreements" minOccurs="0"/>
    <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
  </xs:sequence>
  <xs:attribute name="nym" type="nymType">
    <xs:annotation>
      <xs:documentation>
        This attribute indicates whether or not the
        Identification mechanisms allow the actions of the Principal to
        be linked to an actual end user.
      </xs:documentation>
    </xs:annotation>
  </xs:attribute>
</xs:complexType>

<xs:complexType name="TechnicalProtectionBaseType">
  <xs:sequence>
    <xs:choice minOccurs="0">
      <xs:element ref="PrivateKeyProtection"/>
      <xs:element ref="SecretKeyProtection"/>
    </xs:choice>
    <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>

<xs:complexType name="OperationalProtectionType">
  <xs:sequence>
    <xs:element ref="SecurityAudit" minOccurs="0"/>
    <xs:element ref="DeactivationCallCenter" minOccurs="0"/>
    <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>

<xs:complexType name="AuthnMethodBaseType">
  <xs:sequence>
    <xs:element ref="PrincipalAuthenticationMechanism" minOccurs="0"/>
    <xs:element ref="Authenticator" minOccurs="0"/>
    <xs:element ref="AuthenticatorTransportProtocol" minOccurs="0"/>
    <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>

<xs:complexType name="GoverningAgreementsType">
  <xs:sequence>
    <xs:element ref="GoverningAgreementRef" maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>

<xs:complexType name="GoverningAgreementRefType">
  <xs:attribute name="governingAgreementRef" type="xs:anyURI" use="required"/>
</xs:complexType>

<xs:complexType name="PrincipalAuthenticationMechanismType">
  <xs:sequence>
    <xs:element ref="Password" minOccurs="0"/>
    <xs:element ref="RestrictedPassword" minOccurs="0"/>
    <xs:element ref="Token" minOccurs="0"/>
    <xs:element ref="Smartcard" minOccurs="0"/>
    <xs:element ref="ActivationPin" minOccurs="0"/>
    <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
  </xs:sequence>
  <xs:attribute name="preauth" type="xs:integer" use="optional"/>
</xs:complexType>
```

```
<xs:group name="AuthenticatorChoiceGroup">
  <xs:choice>
    <xs:element ref="PreviousSession"/>
    <xs:element ref="ResumeSession"/>
    <xs:element ref="DigSig"/>
    <xs:element ref="Password"/>
    <xs:element ref="RestrictedPassword"/>
    <xs:element ref="ZeroKnowledge"/>
    <xs:element ref="SharedSecretChallengeResponse"/>
    <xs:element ref="SharedSecretDynamicPlaintext"/>
    <xs:element ref="IPAddress"/>
    <xs:element ref="AsymmetricDecryption"/>
    <xs:element ref="AsymmetricKeyAgreement"/>
    <xs:element ref="SubscriberLineNumber"/>
    <xs:element ref="UserSuffix"/>
    <xs:element ref="ComplexAuthenticator"/>
  </xs:choice>
</xs:group>

<xs:group name="AuthenticatorSequenceGroup">
  <xs:sequence>
    <xs:element ref="PreviousSession" minOccurs="0"/>
    <xs:element ref="ResumeSession" minOccurs="0"/>
    <xs:element ref="DigSig" minOccurs="0"/>
    <xs:element ref="Password" minOccurs="0"/>
    <xs:element ref="RestrictedPassword" minOccurs="0"/>
    <xs:element ref="ZeroKnowledge" minOccurs="0"/>
    <xs:element ref="SharedSecretChallengeResponse" minOccurs="0"/>
    <xs:element ref="SharedSecretDynamicPlaintext" minOccurs="0"/>
    <xs:element ref="IPAddress" minOccurs="0"/>
    <xs:element ref="AsymmetricDecryption" minOccurs="0"/>
    <xs:element ref="AsymmetricKeyAgreement" minOccurs="0"/>
    <xs:element ref="SubscriberLineNumber" minOccurs="0"/>
    <xs:element ref="UserSuffix" minOccurs="0"/>
    <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
  </xs:sequence>
</xs:group>

<xs:complexType name="AuthenticatorBaseType">
  <xs:sequence>
    <xs:group ref="AuthenticatorChoiceGroup"/>
    <xs:group ref="AuthenticatorSequenceGroup"/>
  </xs:sequence>
</xs:complexType>

<xs:complexType name="ComplexAuthenticatorType">
  <xs:sequence>
    <xs:group ref="AuthenticatorChoiceGroup"/>
    <xs:group ref="AuthenticatorSequenceGroup"/>
  </xs:sequence>
</xs:complexType>

<xs:complexType name="AuthenticatorTransportProtocolType">
  <xs:sequence>
    <xs:choice minOccurs="0">
      <xs:element ref="HTTP"/>
      <xs:element ref="SSL"/>
      <xs:element ref="MobileNetworkNoEncryption"/>
      <xs:element ref="MobileNetworkRadioEncryption"/>
      <xs:element ref="MobileNetworkEndToEndEncryption"/>
      <xs:element ref="WTLS"/>
      <xs:element ref="IPSec"/>
      <xs:element ref="PSTN"/>
      <xs:element ref="ISDN"/>
      <xs:element ref="ADSL"/>
    </xs:choice>
    <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>
```

```
  <xs:complexType name="KeyActivationType">
    <xs:sequence>
      <xs:element ref="ActivationPin" minOccurs="0"/>
      <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>

  <xs:complexType name="KeySharingType">
    <xs:attribute name="sharing" type="xs:boolean" use="required"/>
  </xs:complexType>

  <xs:complexType name="PrivateKeyProtectionType">
    <xs:sequence>
      <xs:element ref="KeyActivation" minOccurs="0"/>
      <xs:element ref="KeyStorage" minOccurs="0"/>
      <xs:element ref="KeySharing" minOccurs="0"/>
      <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>

  <xs:complexType name="PasswordType">
    <xs:sequence>
      <xs:element ref="Length" minOccurs="0"/>
      <xs:element ref="Alphabet" minOccurs="0"/>
      <xs:element ref="Generation" minOccurs="0"/>
      <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:attribute name="ExternalVerification" type="xs:anyURI" use="optional"/>
  </xs:complexType>

  <xs:element name="RestrictedPassword" type="RestrictedPasswordType"/>

  <xs:complexType name="RestrictedPasswordType">
    <xs:complexContent>
      <xs:restriction base="PasswordType">
        <xs:sequence>
          <xs:element name="Length" type="RestrictedLengthType" minOccurs="1"/>
          <xs:element ref="Generation" minOccurs="0"/>
          <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
        </xs:sequence>
        <xs:attribute name="ExternalVerification" type="xs:anyURI"
use="optional"/>
      </xs:restriction>
    </xs:complexContent>
  </xs:complexType>

  <xs:complexType name="RestrictedLengthType">
    <xs:complexContent>
      <xs:restriction base="LengthType">
        <xs:attribute name="min" use="required">
          <xs:simpleType>
            <xs:restriction base="xs:integer">
              <xs:minInclusive value="3"/>
            </xs:restriction>
          </xs:simpleType>
        </xs:attribute>
        <xs:attribute name="max" type="xs:integer" use="optional"/>
      </xs:restriction>
    </xs:complexContent>
  </xs:complexType>

  <xs:complexType name="ActivationPinType">
    <xs:sequence>
      <xs:element ref="Length" minOccurs="0"/>
      <xs:element ref="Alphabet" minOccurs="0"/>
      <xs:element ref="Generation" minOccurs="0"/>
      <xs:element ref="ActivationLimit" minOccurs="0"/>
      <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
```

```
<xs:element name="Alphabet" type="AlphabetType"/>
<xs:complexType name="AlphabetType">
  <xs:attribute name="requiredChars" type="xs:string" use="required"/>
  <xs:attribute name="excludedChars" type="xs:string" use="optional"/>
  <xs:attribute name="case" type="xs:string" use="optional"/>
</xs:complexType>

<xs:complexType name="TokenType">
  <xs:sequence>
    <xs:element ref="TimeSyncToken"/>
    <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>

<xs:simpleType name="DeviceTypeType">
  <xs:restriction base="xs:NMTOKEN">
    <xs:enumeration value="hardware"/>
    <xs:enumeration value="software"/>
  </xs:restriction>
</xs:simpleType>

<xs:simpleType name="booleanType">
  <xs:restriction base="xs:NMTOKEN">
    <xs:enumeration value="true"/>
    <xs:enumeration value="false"/>
  </xs:restriction>
</xs:simpleType>

<xs:complexType name="TimeSyncTokenType">
  <xs:attribute name="DeviceType" type="DeviceTypeType" use="required"/>
  <xs:attribute name="SeedLength" type="xs:integer" use="required"/>
  <xs:attribute name="DeviceInHand" type="booleanType" use="required"/>
</xs:complexType>

<xs:complexType name="ActivationLimitType">
  <xs:choice>
    <xs:element ref="ActivationLimitDuration"/>
    <xs:element ref="ActivationLimitUsages"/>
    <xs:element ref="ActivationLimitSession"/>
  </xs:choice>
</xs:complexType>

<xs:element name="ActivationLimitDuration" type="ActivationLimitDurationType">
  <xs:annotation>
    <xs:documentation>
      This element indicates that the Key Activation Limit is
      defined as a specific duration of time.
    </xs:documentation>
  </xs:annotation>
</xs:element>

<xs:element name="ActivationLimitUsages" type="ActivationLimitUsagesType">
  <xs:annotation>
    <xs:documentation>
      This element indicates that the Key Activation Limit is
      defined as a number of usages.
    </xs:documentation>
  </xs:annotation>
</xs:element>

<xs:element name="ActivationLimitSession" type="ActivationLimitSessionType">
  <xs:annotation>
    <xs:documentation>
      This element indicates that the Key Activation Limit is
      the session.
    </xs:documentation>
  </xs:annotation>
</xs:element>

<xs:complexType name="ActivationLimitDurationType">
  <xs:attribute name="duration" type="xs:duration" use="required"/>
```

```
  </xs:complexType>

  <xs:complexType name="ActivationLimitUsagesType">
    <xs:attribute name="number" type="xs:integer" use="required"/>
  </xs:complexType>

  <xs:complexType name="ActivationLimitSessionType"/>

  <xs:complexType name="LengthType">
    <xs:attribute name="min" type="xs:integer" use="required"/>
    <xs:attribute name="max" type="xs:integer" use="optional"/>
  </xs:complexType>

  <xs:simpleType name="mediumType">
    <xs:restriction base="xs:NMTOKEN">
      <xs:enumeration value="memory"/>
      <xs:enumeration value="smartcard"/>
      <xs:enumeration value="token"/>
      <xs:enumeration value="MobileDevice"/>
      <xs:enumeration value="MobileAuthCard"/>
    </xs:restriction>
  </xs:simpleType>

  <xs:complexType name="KeyStorageType">
    <xs:attribute name="medium" type="mediumType" use="required"/>
  </xs:complexType>

  <xs:complexType name="SecretKeyProtectionType">
    <xs:sequence>
      <xs:element ref="KeyActivation" minOccurs="0"/>
      <xs:element ref="KeyStorage" minOccurs="0"/>
      <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>

  <xs:complexType name="SecurityAuditType">
    <xs:sequence>
      <xs:element ref="SwitchAudit" minOccurs="0"/>
      <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>

  <xs:complexType name="ExtensionOnlyType">
    <xs:sequence>
      <xs:element ref="Extension" minOccurs="0"  maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>

  <xs:element name="Extension" type="ExtensionType"/>

  <xs:complexType name="ExtensionType">
    <xs:sequence>
      <xs:any namespace="##other" processContents="lax" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>

</xs:schema>


<?xml version="1.0" encoding="UTF-8"?>
<xs:schema
  targetNamespace="urn:oasis:names:tc:SAML:2.0:ac"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns="urn:oasis:names:tc:SAML:2.0:ac"
  blockDefault="substitution"
  version="2.0">

  <xs:annotation>
    <xs:documentation>
      Document identifier: saml-schema-authn-context-2.0
      Location: http://docs.oasis-open.org/security/saml/v2.0/
```

```
      Revision history:
        V2.0 (March, 2005):
          New core authentication context schema for SAML V2.0.
          This is just an include of all types from the schema
          referred to in the include statement below.
    </xs:documentation>
  </xs:annotation>

  <xs:include schemaLocation="saml-schema-authn-context-types-2.0.xsd"/>

</xs:schem
```

NOTE – L'utilisation de SSL est présentée à l'Appendice IV.

Appendice VII

Profil d'attribut PAC de DCE de SAML

Le présent appendice expose le profil de liaison SAML pour l'environnement de calcul distribué (DCE, *distributed computing environment*), certificats d'attribut priviligié (voir DCE opensource).

VII.1 Profil d'atribut PAC de DCE

Le profil d'attribut PAC de DCE définit l'expression des informations PAC de DCE comme noms et valeurs d'attributs SAML. Il est utilisé pour normaliser un mappage entre les informations principales qui constituent l'identité d'un principal de DCE et un ensemble d'attributs SAML. Ce profil est construit sur le profil d'attribut UUID défini au paragraphe 11.4.9.3.

1) Informations requises

- **Identification**: `urn:oasis:names:tc:SAML:2.0:profiles:attribute:DCE` (c'est aussi l'espace de nom cible alloué au schéma de profil d'attribut PAC de DCE correspondant dans l'Annexe A)
- **Informations de contact**: security-services-comment@lists.oasis-open.org
- **Description**: donnée ci-dessous.
- **Mises à jour**: aucune.

2) Description du PAC

Un PAC DCE est une structure extensible qui peut porter des attributs arbitraires de registre de DCE, mais un ensemble central d'informations est commun entre les principaux et constitue la trame d'une identité de DCE:

- le "domaine" ou "cellule" DEC du principal;
- l'identifiant unique du principal;
- le plus important groupe local DCE dont le principal est membre;
- l'ensemble des groupes locaux DCE dont le principal est membre (plusieurs valeurs);
- l'ensemble des groupes étrangers DCE dont le principal est membre (plusieurs valeurs);

La ou les principales valeurs de chacun de ces attributs est un UUID.

3) Dénomination d'attribut SAML

Ce profil définit un mappage des informations de DCE spécifiques dans les attributs SAML, et donc définit les noms des attributs spécifiques réels, plutôt qu'une convention de dénomination.

Pour tous les attributs définis par ce profil, l'attribut `NameFormat` XML dans les éléments `<Attribute>` doit avoir la valeur urn:`oasis:names:tc:SAML:2.0:attrname-format:uri`.

Pour permettre la lecture par l'homme, il peut aussi y avoir une exigence de certaines applications de porter une chaîne de nom facultative avec l'URI. L'attribut XML facultatif `FriendlyName` peut être utilisée à cette fin.

4) Comparaison de nom d'attribut

Deux éléments `<Attribute>` se réfèrent au même attribut SAML si et seulement si leurs valeurs d'attribut XML Name sont égales, au sens de la Rec. UIT-T X.667. L'attribut `FriendlyName` ne joue aucun rôle dans la comparaison.

5) Attributs XML spécifiques du profil

Aucun attribut XML spécifique n'est défini pour être utilisé avec l'élément `<Attribute>`.

6) Valeurs d'attribut SAML

La ou les principales valeurs de chacun des attributs définis par ce profil sont des UUID. La syntaxe d'URN décrite au § 11.4.9.3 est utilisée pour représenter de telles valeurs.

Cependant, des informations supplémentaires associées à la valeur de l'UUID sont permises par ce profil, consistant en une chaîne facilement lisible par l'homme, et d'un UUID supplémentaire représentant une cellule ou domaine DCE. Des informations supplémentaires sont portées dans l'élément `<AttributeValue>` dans les attributs XML `FriendlyName` et Realm définis par l'espace de nom XML `urn:oasis:names:tc:SAML:2.0:profiles:attribute:DCE`. Ce n'est pas le même que l'attribut XML `FriendlyName` défini au § 8, bien qu'il ait le même objet de base.

La liste du schéma qui suit montre comment utiliser les attributs et types complexes XML spécifiques du profil dans un `xsi:type` [Annexe A]:

```
<schema
targetNamespace="urn:oasis:names:tc:SAML:2.0:profiles:attribute:DCE"
    xmlns:dce="urn:oasis:names:tc:SAML:2.0:profiles:attribute:DCE"
    xmlns="http://www.w3.org/2001/XMLSchema"
    elementFormDefault="unqualified"
    attributeFormDefault="unqualified"
    blockDefault="substitution"
    version="2.0">
    <annotation>
        <documentation>
            Document identifier: saml-schema-dce-2.0
            Location: http://docs.oasis-open.org/security/saml/v2.0/
            Revision history:
            V2.0 (March, 2005):
                Custom schema for DCE attribute profile, first published in
SAML 2.0.
        </documentation>
    </annotation>
    <complexType name="DCEValueType">
        <simpleContent>
            <extension base="anyURI">
                <attribute ref="dce:Realm" use="optional"/>
                <attribute ref="dce:FriendlyName" use="optional"/>
            </extension>
        </simpleContent>
    </complexType>
    <attribute name="Realm" type="anyURI"/>
    <attribute name="FriendlyName" type="string"/>
</schema>
```

7) Définitions d'attribut

Ci-après figure l'ensemble des attributs SAML définis par ce profil. Dans chaque cas, un attribut XML `xsi:type` peut être inclus dans l'élément `<AttributeValue>`, mais doit avoir la valeur **dce:DCEValueType**, où le préfixe dce est arbitraire et doit être lié à l'espace de nom XML `urn:oasis:names:tc:SAML:2.0:profiles:attribute:DCE`.

Une telle utilisation de xsi:type requiert des consommateurs qui valident les attributs d'inclure le schéma d'extension défini par ce profil.

a) Domaine

Cet attribut à valeur unique représente le domaine ou la cellule de DEC du sujet de l'assertion SAML.

Nom: urn:oasis:names:tc:SAML:2.0:profiles:attribute:DCE:realm

L'élément unique `<AttributeValue>` contient un UUID en forme d'URN qui identifie le domaine/cellule de DEC du sujet de l'assertion SAML, avec un attribut XML facultatif `FriendlyName` spécifique du profil qui contient le nom de chaîne du domaine.

b) Principal

Cet attribut à valeur unique représente l'identité principale de DCE du sujet de l'assertion SAML.

Nom: urn:oasis:names:tc:SAML:2.0:profiles:attribute:DCE:principal

L'élément unique `<AttributeValue>` contient un UUID en forme d'URN qui identifie l'identité principale de DCE du sujet de l'assertion SAML, avec un attribut XML `FriendlyName` facultatif spécifique du profil qui contient le nom de chaîne du principal.

L'attribut XML Realm spécifique du profil peut être inclus et doit contenir un UUID en forme d'URN qui identifie le domaine/cellule de DCE du sujet de l'assertion SAML.

c) Groupe principal

L'attribut à une seule valeur représente le groupe de DCE principal dont est membre le sujet de l'assertion SAML.

Nom: urn:oasis:names:tc:SAML:2.0:profiles:attribute:DCE:primary-group

L'élément `<AttributeValue>` unique contient un UUID en forme d'URN qui identifie le groupe principal de DCE du sujet de l'assertion SAML, avec un attribut XML facultatif `FriendlyName` spécifique du profil contenant le nom de chaîne du groupe.

L'attribut XML Realm spécifique du profil peut être inclus et doit contenir un UUID en forme d'URN identifiant le domaine/cellule de DCE du sujet de l'assertion SAML.

d) Groupes

Cet attribut multi-valeurs représente le groupe local de DCE dont le sujet de l'assertion SALM est membre.

Nom: urn:oasis:names:tc:SAML:2.0:profiles:attribute:DCE:groups

Chaque élément `<AttributeValue>` contient un UUID en forme d'URN identifiant un membre du groupe de DCE du sujet de l'assertion SAML, avec un attribut XML facultatif `FriendlyName` spécifique du profil contenant le nom de chaîne du groupe.

L'attribut XML Realm spécifique du profil peut être inclus et doit contenir un UUID en forme d'URN identifiant le domaine/cellule de DCE du sujet de l'assertion SAML.

e) Groupes étrangers

Cet attribut multi-valeurs représente le groupe étranger de DCE dont le sujet de l'assertion SALM est membre.

Nom: urn:oasis:names:tc:SAML:2.0:profiles:attribute:DCE:foreign-groups

Chaque élément `<AttributeValue>` contient un UUID en forme d'URN identifiant un membre du groupe étranger de DCE du sujet de l'assertion SAML, avec un attribut XML facultatif `FriendlyName` spécifique du profil contenant le nom de chaîne du groupe.

L'attribut XML Realm spécifique du profil doit être inclus et doit contenir un UUID en forme d'URN identifiant le domaine/cellule de DCE du groupe étranger.

VII.2 Schéma de DCE SAML

Ci-après figure le schéma de contexte d'authentification SAML pour l'environnement de calcul distribué (DCE).

```
<?xml version="1.0" encoding="UTF-8"?>
<schema targetNamespace="urn:oasis:names:tc:SAML:2.0:profiles:attribute:DCE"
    xmlns:dce="urn:oasis:names:tc:SAML:2.0:profiles:attribute:DCE"
    xmlns="http://www.w3.org/2001/XMLSchema"
    elementFormDefault="unqualified"
    attributeFormDefault="unqualified"
    blockDefault="substitution"
    version="2.0">
    <annotation>
        <documentation>
            Document identifier: saml-schema-dce-2.0
            Location: http://docs.oasis-open.org/security/saml/v2.0/
            Revision history:
```

```
            V2.0 (March, 2005):
                Custom schema for DCE attribute profile, first published in SAML 2.0.
        </documentation>
    </annotation>
    <complexType name="DCEValueType">
        <simpleContent>
            <extension base="anyURI">
                <attribute ref="dce:Realm" use="optional"/>
                <attribute ref="dce:FriendlyName" use="optional"/>
            </extension>
        </simpleContent>
    </complexType>
    <attribute name="Realm" type="anyURI"/>
    <attribute name="FriendlyName" type="string"/>
</schema>
```

VII.3 Exemple

Ci-après figure un exemple de la transformation de données de PAC en attributs SAML appartenant à un principal de DCE nommé "jdoe" dans le domaine "example.com", membre des groupes locaux "cubicle-dwellers" et "underpaid" let d'un groupe étranger "engineers".

```
<saml:Assertion
xmlns:dce="urn:oasis:names:tc:SAML:2.0:profiles:attribute:DCE" ...>
  <saml:Issuer>...</saml:Issuer>
  <saml:Subject>...</saml:Subject>
  <saml:AttributeStatement>
  <saml:Attribute NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-
format:uri"
       Name="urn:oasis:names:tc:SAML:2.0:profiles:attribute:DCE:realm">
    <saml:AttributeValue xsi:type="dce:DCEValueType"
dce:FriendlyName="example.com">
    urn:uuid:003c6cc1-9ff8-10f9-990f-004005b13a2b
    </saml:AttributeValue>
  </saml:Attribute>
  <saml:Attribute NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-
format:uri"
       Name="urn:oasis:names:tc:SAML:2.0:profiles:attribute:DCE:principal">
    <saml:AttributeValue xsi:type="dce:DCEValueType"
dce:FriendlyName="jdoe">
    urn:uuid:00305ed1-a1bd-10f9-a2d0-004005b13a2b
    </saml:AttributeValue>
  </saml:Attribute>
  <saml:Attribute NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-
format:uri"
       Name="urn:oasis:names:tc:SAML:2.0:profiles:attribute:DCE:primary-
group">
    <saml:AttributeValue xsi:type="dce:DCEValueType"
       dce:FriendlyName="cubicle-dwellers">
    urn:uuid:008c6181-a288-10f9-b6d6-004005b13a2b
    </saml:AttributeValue>
  </saml:Attribute>
  <saml:Attribute NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-
format:uri"
       Name="urn:oasis:names:tc:SAML:2.0:profiles:attribute:DCE:groups">
    <saml:AttributeValue xsi:type="dce:DCEValueType"
       dce:FriendlyName="cubicle-dwellers">
    urn:uuid:008c6181-a288-10f9-b6d6-004005b13a2b
    </saml:AttributeValue>
    <saml:AttributeValue xsi:type="dce:DCEValueType"
dce:FriendlyName="underpaid">
    urn:uuid:006a5a91-a2b7-10f9-824d-004005b13a2b
    </saml:AttributeValue>
  </saml:Attribute>
  <saml:Attribute NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-
format:uri"
       Name="urn:oasis:names:tc:SAML:2.0:profiles:attribute:DCE:foreign-
groups">
```

```
    <saml:AttributeValue xsi:type="dce:DCEValueType"
dce:FriendlyName="engineers"
       dce:Realm="urn:uuid:00583221-a35f-10f9-8b6e-004005b13a2b">
    urn:uuid:00099cf1-a355-10f9-9e95-004005b13a2b
    </saml:AttributeValue>
  </saml:Attribute>
  </saml:AttributeStatement>
</saml:Assertion>
```

Appendice VIII

Précisions d'OASIS sur SAML

Le présent appendice ajoute les révisions faites sur SAML v2.0 au sein d'OASIS. Le groupe SAML d'OASIS a décidé de publier ces commentaires d'éclaircissements comme document séparé (voir OASIS:2006 PE). Ces éclaircissements ne sont pas normatifs et n'ont pas été incorporés à la version 2.0 de SAML d'OASIS. Dans la présente Recommandation, ces révisons figurent dans cet Appendice pour que ceux qui mettent en œuvre SAML soient au courant des discussions qui ont eu lieu après la publication de OASIS SAML v2.0 comme norme OASIS.

VIII.1 Erratum potentiel: PE14

Description: Allowcreate doit être défini plus clairement

Applicabilité dans la Recommandation:

Se reporter aux Notes appropriées des § 8.2.4.1 et 8.2.6. De plus, au § 8.2.6.3, une précision du second alinéa du paragraphe est fournie ci-dessous:

Si l'élément `<Terminate>` est inclus dans la demande, le fournisseur demandeur indique que (dans le cas d'un fournisseur de service) il n'acceptera plus d'assertions de la part du fournisseur d'identité ou (dans le cas d'un fournisseur d'identité) il ne produira plus d'assertions au fournisseur de service sur le principal.

Si le fournisseur receveur maintient l'état associé à l'identifiant de nom, comme la valeur de l'identifiant lui-même (dans le cas d'un identifiant en forme de paire), une valeur `SPProvidedID`, le consentement de l'envoyeur à la création/utilisation de l'identifiant, etc., le receveur peut alors effectuer une maintenance quelconque en sachant que la relation représentée par l'identifiant de nom est terminée.

Toute opération ultérieure effectuée par le receveur au nom de l'envoyeur concernant le principal (par exemple, un `<AuthnRequest>` ultérieur) devrait être faite de manière cohérente avec l'absence de tout état antérieur.

La terminaison est potentiellement l'étape de nettoyage de tout comportement de gestion d'état déclenché par l'utilisation de l'attribut `AllowCreate` dans le protocole de demande d'authentification du § 8.2.4. Les développements qui n'utilisent pas cet attribut éviteront vraisemblablement l'utilisation de l'élément `<Terminate>` ou le traiteront comme un élément purement informatif.

Noter que dans la plupart des cas (une exception notable étant les règles qui entourent l'attribut `SPProvidedID`), il n'y a pas d'exigence sur les fournisseurs d'identité ni sur les fournisseurs de service en ce qui concerne la création ou l'utilisation d'un état persistant. Donc, aucun comportement explicite n'est obligatoire quand l'élément `<Terminate>` reçu est 450. Cependant, si un état persistant est présent qui relève de l'utilisation d'un identifiant (comme si un attribut `SPProvidedID` est attaché), l'élément `<Terminate>` fournit une indication claire que cet état devrait être supprimé (ou marqué comme obsolète d'une façon quelconque)."

VIII.2 Erratum potentiel: PE26

Description: le profil SSO doit être précisé

Applicabilité dans la Recommandation: les paragraphes suivants sont précisés comme suit:

11.4.1.4.2 Utilisation de <Response>

Si le fournisseur d'identité souhaite retourner une erreur, il ne doit pas inclure d'assertion dans le message `<Response>`. Autrement, si la demande est réussie (ou si la réponse n'est pas associée à la demande), l'élément `<Response>` doit être conforme à ce qui suit:

- si la réponse n'est pas signée, l'élément `<Issuer>` peut être omis, mais s'il est présent (ou si la réponse est signée) il doit contenir l'identifiant unique du fournisseur d'identité producteur; l'attribut `Format` doit être omis ou avoir une valeur de `urn:oasis:names:tc:SAML:2.0:nameid-format:entity`;
- il doit contenir au moins un `<Assertion>`. Chaque élément `<Issuer>` d'assertion doit contenir l'identifiant unique du fournisseur d'identité répondant; l'attribut Format doit être omis ou avoir une valeur de `urn:oasis:names:tc:SAML:2.0:nameid-format:entity`. Noter que ce profil suppose un seul fournisseur d'identité répondant, et toutes les assertions dans une réponse doivent être produites par la même entité;
- si plusieurs assertions sont incluses, chaque élément `<Subject>` d'assertion doit se référer au même principal. Il est admis que le contenu des éléments `<Subject>` diffèrent (par exemple, en utilisant différents éléments `<NameID>` ou des `<SubjectConfirmation>` de remplacement);
- toute assertion produite pour la consommation utilisant ce profil doit contenir un élément <Subject> avec au moins un élément `<SubjectConfirmation>` contenant une `Method` de `urn:oasis:names:tc:SAML:2.0:cm:bearer`. Une telle assertion est appelée une assertion de titulaire. Les assertions de titulaire peuvent contenir des éléments `<SubjectConfirmation>` supplémentaires;
- les assertions sans un `<SubjectConfirmation>` de titulaire peuvent aussi être incluses; Le traitement de ces assertions supplémentaires ou des éléments `<SubjectConfirmation>` est en dehors du domaine d'application de ce profil;
- au moins un élément `<SubjectConfirmation>` de titulaire doit contenir un élément `<SubjectConfirmationData>` qui doit lui-même contenir un attribut `Recipient` contenant l'URL du service de consommateur d'assertion du fournisseur de service et un attribut `NotOnOrAfter` qui limite la fenêtre pendant laquelle l'assertion peut être délivrée. Il peut aussi contenir un attribut Address qui limite l'adresse client à partir de laquelle l'assertion peut être délivrée. Il ne doit pas contenir d'attribut `NotBefore`. Si le message contenant est en réponse à une `<AuthnRequest>`, l'attribut `InResponseTo` doit alors correspondre à l'identifiant de la demande;
- l'ensemble de une ou plusieurs assertions de titulaire doit contenir au moins un `<AuthnStatement>` qui reflète l'authentification du principal auprès du fournisseur d'identité. Plusieurs éléments `<AuthnStatement>` peuvent être inclus, mais la sémantique de plusieurs déclarations n'est pas définie dans ce profil;
- si le fournisseur d'identité accepte le profil de fin de session unique, défini au § 11.4.1.4.5, toute déclaration d'authentification doit inclure un attribut `SessionIndex` pour permettre une demande de fin de session session par session par le fournisseur de service;
- d'autres déclarations peuvent être incluses dans la ou les assertions de titulaire à la discrétion du fournisseur d'identité. En particulier, les éléments `<AttributeStatement>` peuvent être inclus. Le `<AuthnRequest>` peut contenir un attribut XML `AttributeConsumingServiceIndex` faisant référence à des informations sur les attributs désirés ou exigés au § 9. Le fournisseur d'identité peut l'ignorer, ou envoyer d'autres attributs à sa discrétion;
- chaque assertion de titulaire doit contenir un `<AudienceRestriction>` incluant l'identifiant unique du fournisseur de service comme un élément `<Audience>`;
- d'autres conditions (et d'autres éléments `<Audience>`) peuvent être inclus à la demande du fournisseur de service ou à la discrétion du fournisseur d'identité. (Bien sûr, de telles conditions doivent toutes être comprises et acceptées par le fournisseur de service afin que l'assertion soit considérée comme valide;
- le fournisseur d'identité n'est pas obligé de satisfaire l'ensemble des `<Conditions>` demandées dans le `<AuthnRequest>`, s'il en est.

11.4.1.4.3 Règles de traitement du message<Response>

Indépendamment de la liaison SAML utilisée, le fournisseur de service doit faire ce qui suit:

- vérifier toutes les signatures présentes sur la ou les assertions ou les réponses;
- vérifier que l'attribut `Recipient` dans le `<SubjectConfirmationData>` du titulaire correspond à l'URL du service consommateur d'assertion auquel le `<Response>` ou son artifice a été délivré;
- vérifier que l'attribut `NotOnOrAfter` dans le `<SubjectConfirmationData>` du titulaire n'est pas dépassé, sous réserve du biais d'horloge admissible entre les fournisseurs;
- vérifier que l'attribut `InResponseTo` dans le `<SubjectConfirmationData>` du titulaire est égal à l'identifiant de son message `<AuthnRequest>` d'origine, sauf si la réponse est non sollicitée, auquel cas l'attribut ne doit pas être présent;
- vérifier que toute assertion sur laquelle on s'appuie est valide par ailleurs. Noter qu'alors que plusieurs éléments `<SubjectConfirmation>` du titulaire peuvent être présents, la réussite de l'évaluation d'un seul de ces éléments conformément à ce profil est suffisante pour confirmer une assertion. Cependant, chaque assertion, si plus d'une est présente, doit être évaluée indépendamment;
- si le `<SubjectConfirmationData>` de titulaire inclut un attribut `Address`, le fournisseur de service peut vérifier l'adresse client de l'agent d'utilisateur par rapport à lui;
- toute assertion qui n'est pas valide, ou dont les exigences de confirmation de sujet ne peuvent pas être satisfaites, devrait être éliminée et ne devrait pas être utilisée pour établir un contexte de sécurité pour le principal;
- si un `<AuthnStatement>` utilisé pour établir un contexte de sécurité pour le principal contient un attribut `SessionNotOnOrAfter`, le contexte de sécurité devrait être éliminé une fois que le temps est écoulé, sauf si le fournisseur de service rétablit l'identité du principal en répétant l'utilisation de ce profil. Noter que si plusieurs éléments `<AuthnStatement>` sont présents, la valeur `SessionNotOnOrAfter` la plus proche de l'heure actuelle devrait être satisfaite.

11.4.1.4.4 Règles de traitement spécifiques de POST

Si la liaison POST HTTP est utilisée pour délivrer le `<Response>`, chaque assertion doit être protégée par une signature numérique. Cela peut être réalisé en signant chaque élément `<Assertion>` individuel ou en signant l'élément `<Response>`.

Le fournisseur de service doit s'assurer que chaque assertion de titulaire n'est pas répétée, en maintenant l'ensemble des valeurs `ID` (d'identifiant) utilisées pendant la durée de validité de l'assertion sur la base de l'attribut `NotOnOrAfter` dans le `<SubjectConfirmationData>`.

BIBLIOGRAPHIE

– **FIPS-197** (2001), *Advanced Encryption Standard* (AES). (Norme de chiffrement évolué).

– **IETF RFC 1738** (1994), *Uniform Resource Locators (URL)*. (Adresses universelles), décembre 1994.

– **IETF RFC 2256** (1997), *A Summary of the X.500 (96) User Schema for use with LDAPv3*. (Résumé du schéma d'utilisateur de X.500 (96) à utiliser avec LDAPv3), décembre 1997.

– **IETF RFC 2279** (1998), *UTF-8, a transformation format of ISO 10646*. (UTF-8, un format de transformation de la norme ISO 10646), janvier 1998.

– **IETF RFC 2743** (2000), *Generic Security Service Application Program Interface Version 2, Update 1*. (Mise à jour 1 de la version 2 de l'interface de programme d'application de service de sécurité générique), janvier 2000.

– **DCE**, *Distributed Computing Environment (DCE)*, Open Source. (Environnement de calcul distribué), Open Source. Voir http://www.opengroup.org/dce.

– **OASIS Authentication Context 2.0**, *Authentication Context for the OASIS Security Assertion Markup Language (SAML) V2.0*, (Contexte d'authentification pour le langage de balisage d'assertion de sécurité (SAML) d'OASIS), 15 mars 2005, http://www.oasis-open.org/apps/org/workgroup/security/.

– **OASIS Bindings 1**, *Bindings and Profiles for the OASIS Security Assertion Markup Language (SAML)*, (Liaisons et profils pour le langage de balisage d'assertion de sécurité (SAML) d'OASIS), 5 novembre 2002, http://www.oasis-open.org/apps/org/workgroup/security/.

– **OASIS Bindings 1.1**, *Bindings and Profiles for the OASIS Security Assertion Markup Language (SAML)*, (Liaisons et profils pour le langage de balisage d'assertion de sécurité (SAML) d'OASIS), 22 septembre 2003, http: http://www.oasis-open.org/apps/org/workgroup/security/.

– **OASIS Bindings 2.0**, *Bindings for the OASIS Security Assertion Markup Language (SAML) V2.0*, 15 mars 2005, http://www.oasis-open.org/apps/org/workgroup/security/.

– **OASIS Conformance 2.0**, *Conformance Requirements for the OASIS Security Assertion Markup Language (SAML) V2.00*, (Exigences de conformité pour le langage de balisage d'assertion de sécurité (SAML) V2.00 d'OASIS), 15 mars 2005, http://www.oasis-open.org/apps/org/workgroup/security/.

– **OASIS Glossary 2.0**, *Glossary for the OASIS Security Assertion Markup Language (SAML) V2.0*, (Glossaire pour le langage de balisage d'assertion de sécurité (SAML) V2.0 d'OASIS), 15 mars 2005, http://www.oasis-open.org/apps/org/workgroup/security/.

– **OASIS Metadata 2.0**, *Metadata for the OASIS Security Assertion Markup Language (SAML) V2.0*, (Métadonnées pour le langage de balisage d'assertion de sécurité (SAML) V2.0 d'OASIS), 15 mars 2005, http://www.oasis-open.org/apps/org/workgroup/security/.

– **OASIS Errata Document 24**, *Revision 24 draft of the non-normative SAML V2.0 Errata document*, (Projet de révision 24 du document d'errata non normatif de SAML V2.0), 27 février 2006, http://www.oasis-open.org/committees/download.php/16935/sstc-saml-errata-2.0-draft-24.pdf.

– **OASIS Protocol 1.0**, *Assertions and Protocol for the OASIS Security Assertion Markup Language (SAML)*, (Assertions et protocoles pour le langage de balisage d'assertion de sécurité (SAML) d'OASIS), 5 novembre 2002, http://www.oasis-open.org/apps/org/workgroup/security/.

– **OASIS Protocol 1.1**, *Assertions and Protocol for the OASIS Security Assertion Markup Language (SAML)*, (Assertions et protocoles pour le langage de balisage d'assertion de sécurité (SAML) d'OASIS), 22 septembre 2003, http://www.oasis-open.org/apps/org/workgroup/security/.

– **OASIS Protocol 2.0**, *Profiles for the OASIS Security Assertion Markup Language (SAML) V2.0*, (Profils pour le langage de balisage d'assertion de sécurité (SAML) V2.0 d'OASIS), 15 mars 2005 http://www.oasis-open.org/apps/org/workgroup/security/.

– **OASIS SAML 1.0**, *Security Assertion Markup Language (SAML) Version 1.0 Specification Set*, (Ensemble de spécification de la Version 1.0 du langage de balisage d'assertion de sécurité (SAML)), 5 novembre 2002, http://www.oasis-open.org/apps/org/workgroup/security/.

– **OASIS SAML 1.1**, *Security Assertion Markup Language (SAML) Version 1.1 Specification Set*, (Ensemble de spécification de la Version 1.1 du langage de balisage d'assertion de sécurité (SAML)), 22 septembre 2003, http://www.oasis-open.org/apps/org/workgroup/security/.

– **OASIS Security 1**, *Security Considerations for the OASIS Security Assertion Markup Language (SAML)*, (Considérations sur la sécurité et la confidentialité pour le langage de balisage d'assertion de sécurité (SAML) d'OASIS), 5 novembre 2002, http://www.oasis-open.org/apps/org/workgroup/security/.

– **OASIS Security 1.1**, *Security Considerations for the OASIS Security Assertion Markup Language (SAML)*, (Considérations sur la sécurité et la confidentialité pour le langage de balisage d'assertion de sécurité (SAML) d'OASIS), 22 septembre 2003, http://www.oasis-open.org/apps/org/workgroup/security/.

– **OASIS Security 2.0**, *Security and Privacy Considerations for the OASIS Security Assertion Markup Language (SAML) V2.0*, (Considérations sur la sécurité et la confidentialité pour le langage de balisage d'assertion de sécurité (SAML) d'OASIS), 15 mars 2005, http://www.oasis-open.org/apps/org/workgroup/security/.

– **OASIS XACML1-1**, *eXtensible Access Control Markup Language (XACML) V1.1*, (Langage de balisage de contrôle d'accès extensible (XACML) V1.1) 24 juillet 2003, http://www.oasis-open.org/apps/org/workgroup/xacml/.

– **OASIS XACML1-1**, *eXtensible Access Control Markup Language (XACML) V1.0*, (Langage de balisage de contrôle d'accès extensible (XACML) V1.0), 18 février 2003, http://www.oasis-open.org/apps/org/workgroup/xacml/.

– **OASIS XACML 2.0**, *eXtensible Access Control Markup Language (XACML) V2.0*, (Langage de balisage de contrôle d'accès extensible (XACML) V2.0) 1 février 2005, http://www.oasis-open.org/apps/org/workgroup/xacml/.

– **SSL3**, *The SSL Protocol Version 3.0*. Voir http://wp.netscape.com/eng/ssl3/draft302.txt

– **W3C Character Model** (2004), Working draft, *Character Model for the World Wide Web 1.0: Normalization.*

SÉRIES DES RECOMMANDATIONS UIT-T

Série A	Organisation du travail de l'UIT-T
Série D	Principes généraux de tarification
Série E	Exploitation générale du réseau, service téléphonique, exploitation des services et facteurs humains
Série F	Services de télécommunication non téléphoniques
Série G	Systèmes et supports de transmission, systèmes et réseaux numériques
Série H	Systèmes audiovisuels et multimédias
Série I	Réseau numérique à intégration de services
Série J	Réseaux câblés et transmission des signaux radiophoniques, télévisuels et autres signaux multimédias
Série K	Protection contre les perturbations
Série L	Construction, installation et protection des câbles et autres éléments des installations extérieures
Série M	Gestion des télécommunications y compris le RGT et maintenance des réseaux
Série N	Maintenance: circuits internationaux de transmission radiophonique et télévisuelle
Série O	Spécifications des appareils de mesure
Série P	Qualité de transmission téléphonique, installations téléphoniques et réseaux locaux
Série Q	Commutation et signalisation
Série R	Transmission télégraphique
Série S	Equipements terminaux de télégraphie
Série T	Terminaux des services télématiques
Série U	Commutation télégraphique
Série V	Communications de données sur le réseau téléphonique
Série X	**Réseaux de données, communication entre systèmes ouverts et sécurité**
Série Y	Infrastructure mondiale de l'information, protocole Internet et réseaux de prochaine génération
Série Z	Langages et aspects généraux logiciels des systèmes de télécommunication

Imprimé en Suisse
Genève, 2007