Union internationale des télécommunications

UIT-T

SECTEUR DE LA NORMALISATION
DES TÉLÉCOMMUNICATIONS
DE L'UIT

Y.2722

(01/2011)

SÉRIE Y: INFRASTRUCTURE MONDIALE DE L'INFORMATION, PROTOCOLE INTERNET ET RÉSEAUX DE PROCHAINE GÉNÉRATION

Réseaux de prochaine génération – Sécurité

Mécanismes de gestion d'identité dans les réseaux de prochaine génération

Recommandation UIT-T Y.2722

RECOMMANDATIONS UIT-T DE LA SÉRIE Y

INFRASTRUCTURE MONDIALE DE L'INFORMATION, PROTOCOLE INTERNET ET RÉSEAUX DE PROCHAINE GÉNÉRATION

INFRASTRUCTURE MONDIALE DE L'INFORMATION	
Généralités	Y.100–Y.199
Services, applications et intergiciels	Y.200–Y.299
Aspects réseau	Y.300–Y.399
Interfaces et protocoles	Y.400–Y.499
Numérotage, adressage et dénomination	Y.500–Y.599
Gestion, exploitation et maintenance	Y.600–Y.699
Sécurité	Y.700–Y.799
Performances	Y.800–Y.899
ASPECTS RELATIFS AU PROTOCOLE INTERNET	
Généralités	Y.1000–Y.1099
Services et applications	Y.1100–Y.1199
Architecture, accès, capacités de réseau et gestion des ressources	Y.1200–Y.1299
Transport	Y.1300–Y.1399
Interfonctionnement	Y.1400–Y.1499
Qualité de service et performances de réseau	Y.1500–Y.1599
Signalisation	Y.1600–Y.1699
Gestion, exploitation et maintenance	Y.1700–Y.1799
Taxation	Y.1800–Y.1899
Télévision IP sur réseaux de prochaine génération	Y.1900–Y.1999
RÉSEAUX DE PROCHAINE GÉNÉRATION	
Cadre général et modèles architecturaux fonctionnels	Y.2000–Y.2099
Qualité de service et performances	Y.2100–Y.2199
Aspects relatifs aux services: capacités et architecture des services	Y.2200–Y.2249
Aspects relatifs aux services: interopérabilité des services et réseaux dans les réseaux de prochaine génération	Y.2250–Y.2299
Numérotage, nommage et adressage	Y.2300–Y.2399
Gestion de réseau	Y.2400–Y.2499
Architectures et protocoles de commande de réseau	Y.2500–Y.2599
Smart ubiquitous networks	Y.2600–Y.2699
Sécurité	**Y.2700–Y.2799**
Mobilité généralisée	Y.2800–Y.2899
Environnement ouvert de qualité opérateur	Y.2900–Y.2999
Réseaux futurs	Y.3000–Y.3099

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

Recommandation UIT-T Y.2722

Mécanismes de gestion d'identité dans les réseaux de prochaine génération

Résumé

La Recommandation UIT-T Y.2722 spécifie les mécanismes pouvant être utilisés pour satisfaire aux spécifications de la gestion d'identité (IdM, *identity management*) et aux besoins de déploiement des réseaux de prochaine génération (NGN, *next generation network*).

Historique

Edition	Recommandation	Approbation	Commission d'études
1.0	ITU-T Y.2722	2011-01-28	13

Mots clés

Identité fédérée, gestion d'identité, mécanismes de gestion d'identité, réseau de prochaine génération, sécurité.

AVANT-PROPOS

L'Union internationale des télécommunications (UIT) est une institution spécialisée des Nations Unies dans le domaine des télécommunications et des technologies de l'information et de la communication (ICT). Le Secteur de la normalisation des télécommunications (UIT-T) 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 avait 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 de 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

		Page
1	Domaine d'application	5
2	Références	5
3	Définitions	6
4	Abréviations	6
5	Conventions	7
6	Mécanismes et procédures de prise en charge des fonctions IdM	8
6.1	Gestion du cycle de vie	8
6.2	Authentification et garantie d'authentification	8
6.3	Corrélation et lien	28
6.4	Découverte	29
6.5	Communications et échanges d'informations IdM	30
6.6	Protection des informations d'identification personnelle (PII)	33
6.7	Fonctions d'identité fédérée	34
6.8	Contrôle d'accès aux informations d'identité	35
6.9	Authentification unique	35
6.10	Déconnexion unique	36
7	Sécurité	39
Appendice I	Authentification de message WSS UIT-T X.509 v3	40
Appendice II	Mécanisme "OpenID + OAuth" pour le contrôle d'accès	42
II.1	OAuth ([b-IETF RFC 5849])	42
II.2	Utilisation de OpenID conjointement avec OAuth	42
II.3	Flux d'autorisation OpenID + OAuth	42
Bibliographie		45

Recommandation UIT-T Y.2722

Mécanismes de gestion d'identité dans les réseaux de prochaine génération

1 Domaine d'application

La Recommandation [UIT-T Y.2721], "*Spécifications et cas d'utilisation de la gestion d'identité dans les NGN*", contient les spécifications de la gestion d'identité applicables aux réseaux de prochaine génération (NGN). La présente Recommandation décrit les mécanismes et séries d'options spécifiques en matière de gestion d'identité qu'il convient d'utiliser pour satisfaire les spécifications énoncées dans [UIT-T Y.2721]. Elle spécifie en outre des bonnes pratiques et des lignes directrices pour la prise en charge de l'interopérabilité et d'autres besoins.

La présente Recommandation est destinée à être utilisée avec [UIT-T Y.2720] et [UIT-T Y.2721], les concepts architecturaux de base, les spécifications et les cas d'utilisation n'étant pas présentés à nouveau ici.

NOTE – Lors de l'implémentation et de l'utilisation des mécanismes décrits, toutes les lois, réglementations et politiques nationales et régionales applicables doivent être respectées. Certaines réglementations et législations peuvent exiger l'implémentation de mécanismes destinés à protéger les informations d'identification personnelle.

2 Références

Les Recommandations UIT-T et autres références suivantes contiennent des dispositions qui, par suite de la référence qui y est faite, constituent des dispositions de la présente Recommandation. Au moment de la publication, les éditions indiquées étaient en vigueur. Les Recommandations et autres références étant sujettes à révision, les utilisateurs de la présente Recommandation sont invités à rechercher la possibilité d'appliquer les éditions les plus récentes des Recommandations et autres références énumérées ci-dessous. Une liste des Recommandations UIT-T en vigueur est publiée périodiquement. La référence à un document figurant dans la présente Recommandation ne donne pas à ce document, en tant que tel, le statut de Recommandation.

[UIT-T X.509] Recommandation UIT-T X.509 (2005) | ISO/CEI 9594-8:2005, *Technologies de l'information – Interconnexion des systèmes ouverts – L'annuaire: cadre général des certificats de clé publique et d'attribut*.

[UIT-T X.1035] Recommandation UIT-T X.1035 (2007), *Protocole d'échange de clés avec authentification par mot de passe*.

[UIT-T X.1141] Recommandation UIT-T X.1141 (2006), *Langage de balisage d'assertion de sécurité (SAML 2.0)*.

[UIT-T X.1252] Recommandation UIT-T X.1252 (2010), *Termes et définitions de base relatifs à la gestion d'identité*.

[UIT-T Y.2012] Recommandation UIT-T Y.2012 (2006), *Prescriptions fonctionnelles et architecture du réseau de prochaine génération version 1*.

[UIT-T Y.2701] Recommandation UIT-T Y.2701 (2007), *Prescriptions de sécurité des réseaux de prochaine génération de version 1*.

[UIT-T Y.2702] Recommandation UIT-T Y.2702 (2008), *Spécifications d'authentification et d'autorisation pour les réseaux de prochaine génération version 1*.

[UIT-T Y.2704] Recommandation UIT-T Y.2704 (2010), *Mécanismes et procédures de sécurité applicables aux réseaux de prochaine génération*.

[UIT-T Y.2720] Recommandation UIT-T Y.2720 (2009), *Cadre de gestion d'identité dans les NGN*.

[UIT-T Y.2721] Recommandation UIT-T Y.2721 (2010), *Spécifications et cas d'utilisation de la gestion d'identité dans les NGN*.

[3GPP TS 23.228] 3GPP TS 23.228 (en vigueur), *IP Multimedia Subsystem (IMS); Stage 2*.

[ATIS 33102] ATIS.3GPP.33.102V710-2007, *Security Architecture*.

[IETF RFC 2289] IETF RFC 2289 (1998), *A One-Time Password System*.

3 Définitions

La présente Recommandation utilise les termes définis dans [UIT-T Y.2720] et dans [UIT-T X.1252].

Elle adopte en particulier les définitions suivantes extraites de [UIT-T X.1252]:

3.1 fournisseur d'identité (IdP, *identity provider*): voir fournisseur de service d'identité (IdSP).

3.2 fournisseur de service d'identité (IdSP, *identity service provider*): entité qui vérifie, tient à jour, gère et peut créer et attribuer des informations d'identité d'autres entités.

4 Abréviations

La présente Recommandation utilise les abréviations et acronymes suivants:

AKA authentification et accord de clé (*authentication and key agreement*)

ASP fournisseur de service d'application (*application service provider*)

AuC centre d'authentification (*authentication centre*)

AV vecteur d'authentification (*authentication vector*)

BSF fonction de serveur d'amorçage (*bootstrapping server function*)

CK clé de chiffrement (*ciphering key*)

GBA architecture d'amorçage générique (*generic bootstrapping architecture*)

HSS serveur d'abonnés de rattachement (*home subscriber server*)

IdM gestion d'identité (*identity management*)

IdP fournisseur d'identité (*identity provider*)

IdSP fournisseur de service d'identité (*identity service provider*)

IK clé d'intégrité (*integrity key*)

IMPI identité d'utilisateur privé multimédia IP (*IP multimedia private user identity*)

IMPU identité d'utilisateur public multimédia IP (*IP multimedia public user identity*)

IMS sous-système multimédia IP (*IP multimedia subsystem*)

IMSI identité internationale d'abonné mobile (*international mobile subscriber identity*)

ISIM module d'identité d'abonné IMS (*IMS subscriber identity module*)

LDAP protocole simple d'accès à l'annuaire (*lightweight directory access protocol*)

MS station mobile (*mobile station*)

NAF fonction d'application de réseau (*network application function*)

NGN réseau de prochaine génération (*next generation network*)

OASIS Organization for the Advancement of Structured Information Standards

OTP mot de passe à usage unique (*one time password*)

PII information d'identification personnelle (*personally identifiable information*)

PKI infrastructure de clé publique (*public key infrastructure*)

RP partie utilisatrice (*relying party*)

SAML langage de balisage d'assertion de sécurité (*security assertion markup language*)

SIP protocole d'ouverture de session (*session initiation protocol*)

SLF fonction de localisation d'abonné (*subscriber locator function*)

SOAP protocole simple d'accès aux objets (*simple object access protocol*)

SQL langage de requête structuré (*structured query language*)

SSO authentification unique (*single sign-on*)

TVIP télévision utilisant le protocole Internet

UE équipement d'utilisateur (*user equipment*)

UICC carte de circuit intégré universelle (*universal integrated circuit card*)

UMTS système de télécommunications mobiles universelles (*universal mobile telecommunications system*)

USIM module d'identité d'abonné universel (*universal subscriber identifier module*)

WAP protocole d'application hertzienne (*wireless application protocol*)

WSS sécurité des services web (*web services security*)

XML langage de balisage extensible (*extensible markup language*)

XRDS séquence de descripteurs de ressources extensible (*extensible resource descriptor sequence*)

5 Conventions

Dans la présente Recommandation:

L'expression "il est obligatoire" indique une spécification qui doit être strictement suivie et par rapport à laquelle aucun écart n'est permis pour pouvoir déclarer la conformité à la présente Recommandation.

L'expression "il est recommandé" indique une spécification qui est recommandée mais qui n'est pas absolument nécessaire. Cette disposition n'est donc pas indispensable pour déclarer la conformité.

L'expression "il est interdit" indique une spécification qui doit être strictement suivie et par rapport à laquelle aucun écart n'est permis pour pouvoir déclarer la conformité à la présente Recommandation.

L'expression "peut, à titre d'option" indique une spécification optionnelle qui est admissible, sans pour autant être en quoi que ce soit recommandée. Elle ne doit pas être interprétée comme l'obligation pour le fabricant de mettre en œuvre l'option et la possibilité pour l'opérateur de réseau ou le fournisseur de service de l'activer ou non, mais comme la possibilité pour le fabricant de fournir ou non cette option, sans que cela n'ait d'incidence sur la déclaration de conformité.

Dans le corps de la présente Recommandation et dans ses appendices, on trouve parfois les expressions *doit*, *ne doit pas*, *devrait* et *peut*. Celles-ci doivent respectivement être interprétées

comme correspondant aux expressions *il est obligatoire, il est interdit, il est recommandé* et *peut, à titre d'option*. Lorsque ces expressions apparaissent dans un appendice ou dans des parties dans lesquelles il est expressément indiqué qu'elles sont *données à titre d'information*, elles doivent être interprétées comme étant dépourvues d'intention normative.

6 Mécanismes et procédures de prise en charge des fonctions IdM

6.1 Gestion du cycle de vie

Voir [UIT-T Y.2720], *Cadre de gestion d'identité dans les NGN*, pour plus d'informations sur la gestion du cycle de vie des identités.

6.2 Authentification et garantie d'authentification

Le présent paragraphe décrit des mécanismes permettant d'authentifier et de garantir des identités et des informations d'identité. Il fait référence à des mécanismes d'authentification définis ailleurs.

Des mécanismes d'authentification spécifiques tels que l'authentification basée sur des services web, un profil de langage de balisage d'assertion de sécurité (SAML, *security assertion markup language*), l'authentification basée sur un certificat ou l'authentification basée sur un mot de passe (par exemple un mot de passe à usage unique) peuvent être utilisés par le fournisseur IdSP pour certaines applications ou certains services suivant le contexte et le niveau de garantie nécessaire. La ou les méthodes d'authentification sont choisies sur la base du niveau de garantie requis. Le fournisseur IdSP peut demander des informations afin de déterminer les méthodes d'authentification qui satisfont au niveau de garantie requis par le fournisseur de service.

6.2.1 Authentification basée sur un profil SAML de sécurité des services web

6.2.1.1 Assertions SAML

Le langage de balisage d'assertion de sécurité (SAML) [UIT-T X.1141] spécifie le format des assertions qui peuvent être utilisées pour l'échange d'informations de sécurité aux fins de la gestion d'identité. Parmi les fonctions IdM qui peuvent être implémentées en utilisant le langage SAML, on peut citer l'authentification, le partage d'attributs et l'autorisation, qui correspondent à trois types de déclaration à propos du sujet d'une assertion SAML:

- Déclaration d'authentification – Indique que le sujet de l'assertion a été authentifié par un moyen particulier à un instant particulier.
- Déclaration d'attribut – Indique que le sujet de l'assertion est associé aux attributs énumérés.
- Déclaration de décision d'autorisation – Indique que l'accès à une ressource spécifiée a été accordé ou refusé au sujet de l'assertion.

Le contenu d'une assertion SAML peut être décrit à un haut niveau comme suit: l'assertion **A** a été émise à l'instant **t** par l'émetteur **R** concernant le sujet **S**, les conditions **C** étant valables.

Les assertions SAML utilisées pour communiquer des informations d'authentification, d'attribut et d'autorisation sont acheminées dans des messages SOAP simples. Lorsque des messages SOAP sont échangés sur un canal de transport non protégé, il est fortement recommandé d'utiliser une signature XML [b-W3C XML signature] afin de pouvoir vérifier la relation entre le message SOAP et les déclarations des assertions acheminées dans le message. La norme *web services security (WSS): SAML token profile* [b-OASIS SAML token] décrit:

- comment des assertions SAML (également appelées jetons SAML) ou des références à des assertions SAML sont acheminées dans un message SOAP.
- comment une signature XML est utilisée pour rattacher un sujet et les déclarations d'une assertion SAML à un message SOAP.

Un exemple d'utilisation d'un jeton SAML avec un message SOAP élaboré conformément à cette Recommandation est illustré sur la Figure 1 et décrit ci-dessous.

Dans cet exemple, un message SOAP signé contient une assertion SAML avec une déclaration d'attribut. Sur la base des informations contenues dans cette déclaration, le récepteur pourrait prendre des décisions de contrôle d'accès.

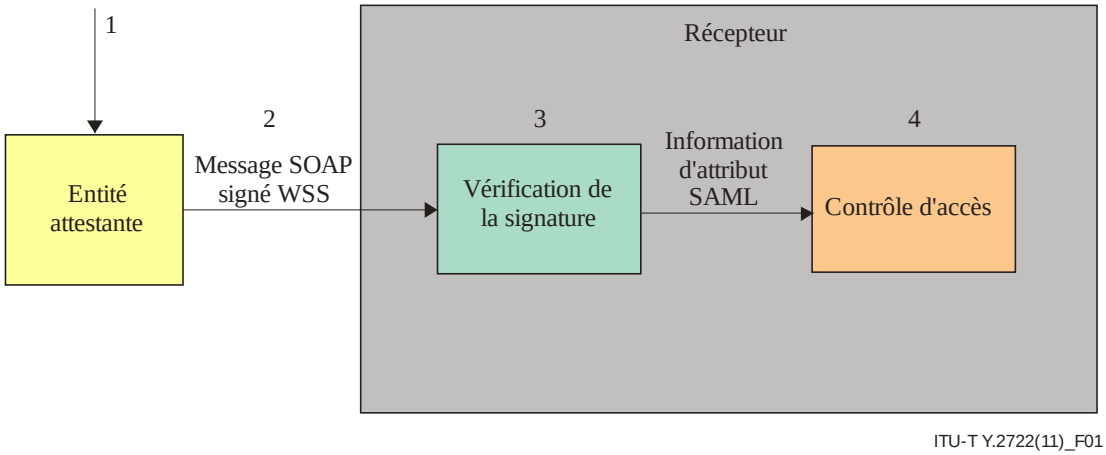


Figure 1 – Etapes types d'élaboration et de traitement d'un message SOAP avec un jeton SAML

1) L'entité attestante obtient une assertion SAML avec une déclaration d'attribut et l'inclut dans un message SOAP conformément à [b-OASIS SAML token].

2) L'entité attestante envoie le message SOAP signé selon la procédure WSS au récepteur.

3) Le récepteur vérifie la signature numérique.

4) Les informations contenues dans la déclaration SAML peuvent être utilisées aux fins de décisions de contrôle d'accès.

6.2.1.2 Méthodes de confirmation de sujet des jetons SAML

La norme OASIS, *Web Services Security: SAML Token Profile 1.1* [b-OASIS SAML token] spécifie comment rattacher une assertion SAML à un message SOAP et définit deux méthodes obligatoires de confirmation du sujet:

- holder-of-key;
- sender-vouches.

Les principaux éléments XML du message SOAP élaboré conformément à [b-OASIS WSS SOAP] sont illustrés sur la Figure 2.

L'assertion SAML est placée dans l'en-tête <wsse:Security>, qui contient également la signature numérique <ds:Signature>. La signature numérique est utilisée par le récepteur du message SOAP pour vérifier que l'expéditeur du message connaît la clé utilisée pour calculer la signature sur le condensé du corps du message SOAP et pour vérifier son intégrité. L'algorithme utilisé pour le condensé est SHA 1 et celui utilisé pour la signature est RSA_SHA 1, comme spécifié dans [b-OASIS WSS SOAP]. La valeur de la signature est fournie dans l'élément <ds:SignatureValue> de la signature numérique <ds:Signature>.

Les deux méthodes de confirmation du sujet définissent des solutions différentes pour acheminer au récepteur des informations relatives sur la clé.

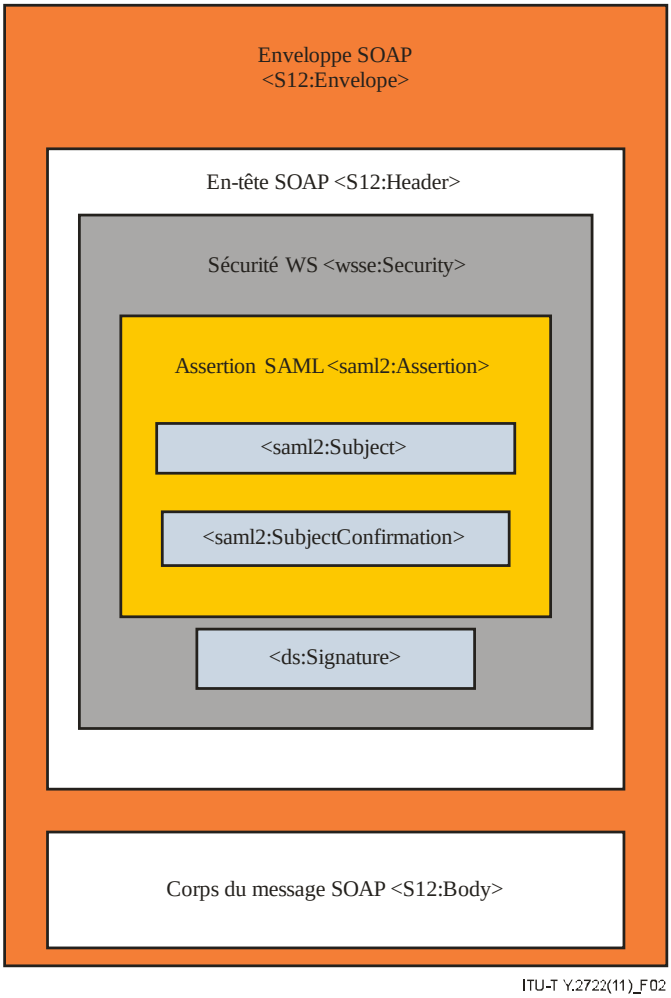


Figure 2 – Structure du message SOAP avec assertion SAML

Les paragraphes qui suivent décrivent les deux méthodes de confirmation du sujet.

6.2.1.2.1 Méthode de confirmation du sujet holder-of-key

La Figure 3 illustre la structure de l'assertion SAML utilisée pour la méthode de confirmation du sujet holder-of-key. L'attribut de méthode de l'élément <saml2:SubjectConfirmation> indique la méthode de confirmation du sujet (holder-of-key).

La méthode spécifie que l'expéditeur (également appelé entité attestante) doit prouver qu'il est habilité à faire des déclarations à propos du sujet en démontrant qu'il connaît la clé, qui est identifiée dans l'élément <ds:KeyValue> contenu dans l'élément <ds:KeyInfo> de l'assertion SAML. L'élément <ds:KeyInfo> identifie une clé publique ou secrète qui est utilisée pour confirmer l'identité du sujet. La méthode spécifie en outre que, pour apporter cette preuve, l'expéditeur peut signer un condensé du corps du message SOAP avec cette clé. La signature est contenue dans l'élément <ds:Signature> de l'en-tête de sécurité des services web comme indiqué sur la Figure 2.

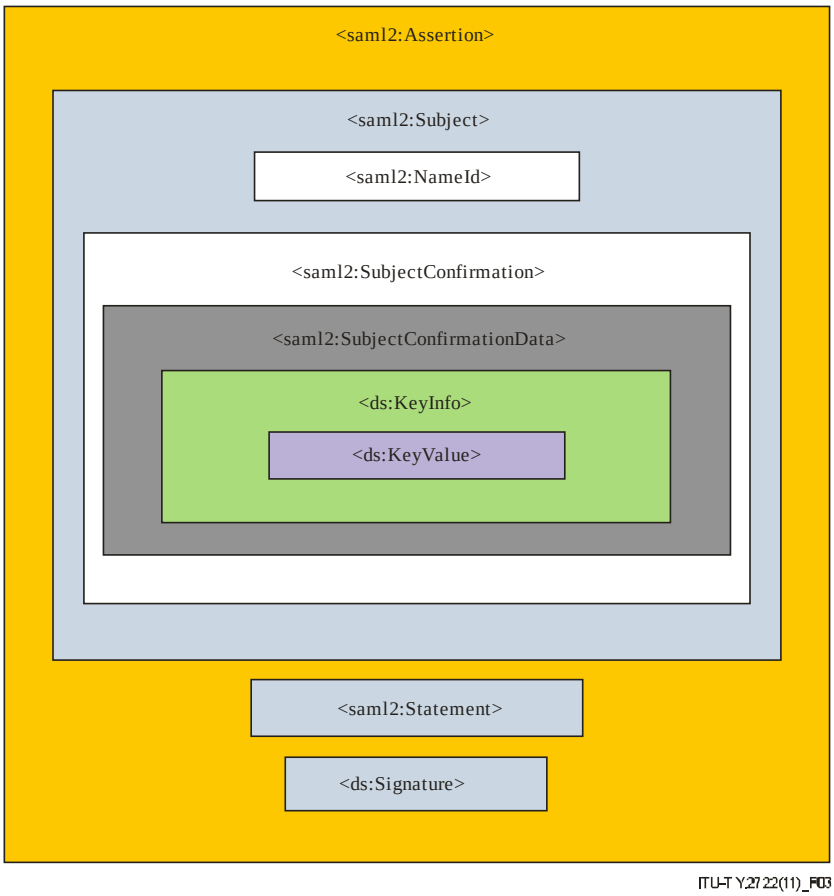


Figure 3 – Structure de l'assertion SAML utilisée pour la méthode de confirmation du sujet holder-of-key

Le récepteur du message SOAP obtient la clé en utilisant les informations qui sont fournies par l'entité attestante dans l'élément <ds:KeyInfo>. Il calcule ensuite la signature numérique du corps du message SOAP et vérifie si elle correspond à la signature fournie par l'entité attestante. Si c'est le cas, le sujet et les déclarations de l'assertion SAML peuvent être attribués à l'entité attestante et le contenu du corps du message SOAP dont l'intégrité est protégée par la clé peut être considéré comme étant fourni par l'entité attestante.

6.2.1.2.2 Méthode de confirmation du sujet sender-vouches

La Figure 4 illustre la structure de l'assertion SAML utilisée pour la méthode de confirmation du sujet sender-vouches. L'attribut de méthode de l'élément <saml2:SubjectConfirmation> indique la méthode de confirmation du sujet (sender-vouches).

L'entité attestante a la confiance d'un récepteur pour faire des assertions SAML concernant un sujet à condition que la valeur de l'attribut de méthode de l'élément <SubjectConfirmation> indique la méthode sender-vouches.

L'entité attestante obtient une ou plusieurs assertions ou références à des assertions auprès d'une ou de plusieurs autorités et les inclut dans un message SOAP. Elle calcule ensuite une signature du condensé des assertions SAML et du corps du message SOAP. La signature est contenue dans l'élément <ds:Signature> de l'en-tête de sécurité des services web (voir la Figure 2). L'entité attestante fournit facultativement au récepteur des informations relatives à la clé qui a été utilisée pour calculer la signature. En l'absence de ces informations, le récepteur doit utiliser un autre moyen pour identifier la clé.

Le récepteur vérifie la validité de la signature. Si la signature est valable, le récepteur établit que les déclarations à propos du sujet ont été faites par l'entité attestante.

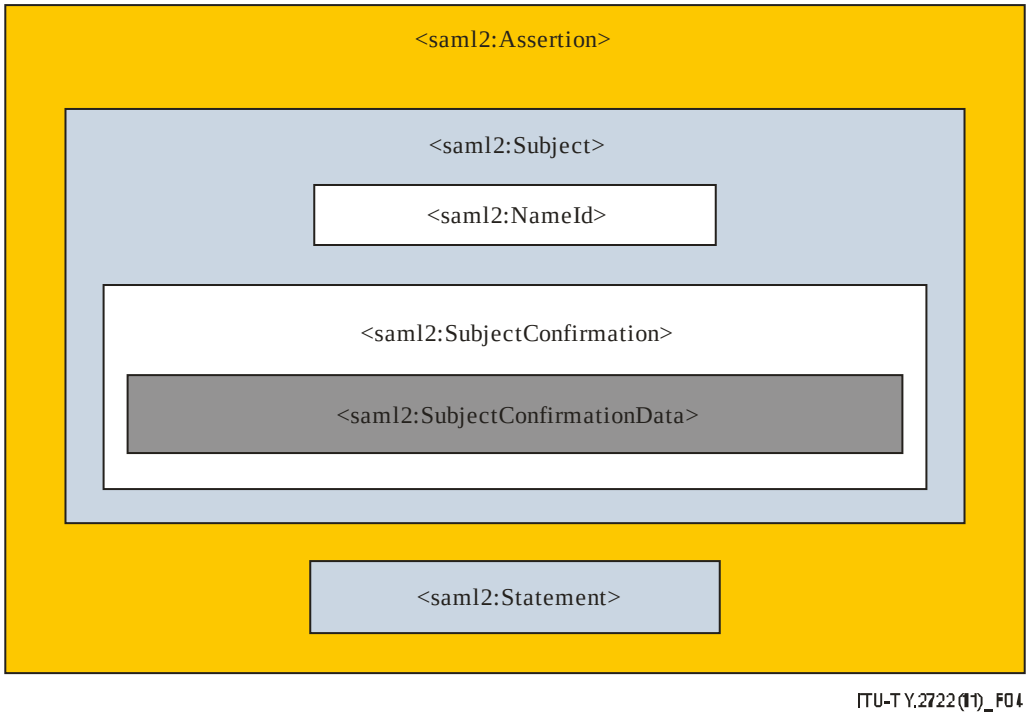


Figure 4 – Structure de l'assertion SAML utilisée pour la méthode de confirmation du sujet sender-vouches

6.2.2 Authentification basée sur un certificat

Des certificats [UIT-T X.509] peuvent être utilisés pour certaines applications ou certains services suivant le contexte et le niveau de garantie nécessaire. L'utilisation de certificats [UIT-T X.509] pour l'authentification est décrite dans [UIT-T Y.2704], *Mécanismes et procédures de sécurité applicables aux réseaux de prochaine génération*.

6.2.3 Authentification basée sur un mot de passe

Une authentification basée sur un mot de passe peut être utilisée pour certaines applications ou certains services suivant le contexte et le niveau de garantie nécessaire. Un mécanisme d'authentification basée sur un mot de passe est décrit dans [UIT-T X.1035].

6.2.4 Mot de passe à usage unique

Un mot de passe à usage unique peut être utilisé pour certaines applications ou certains services suivant le contexte et le niveau de garantie nécessaire. Une méthode d'implémentation du mot de passe à usage unique est décrite dans [IETF RFC 2289].

6.2.5 Utilisation de l'authentification et de l'accord de clé (AKA) pour l'authentification mutuelle

Le protocole d'authentification et d'accord de clé (AKA, *authentication and key agreement*) du système de télécommunications mobiles universelles (UMTS, *universal mobile telecommunications system*) peut être utilisé pour assurer l'authentification mutuelle de la station mobile et du réseau. C'est un protocole de type défi-réponse, reposant sur le partage d'une clé à long terme entre le module d'identité d'abonné universel (USIM, *universal subscriber identity module*) et le centre d'authentification (AuC). Ces entités résident respectivement sur la carte de circuit intégré universelle (UICC, *universal integrated circuit card*) de la station mobile et dans le réseau de rattachement de la station mobile. Dans le cadre de certains arrangements commerciaux, les fonctions de l'AuC pourraient être assurées par un fournisseur IdSP. Le protocole AKA est spécifié dans [ATIS 33102].

6.2.6 Intégration de l'authentification basée sur l'infrastructure PKI avec le sous-système IMS

La sécurité des sous-systèmes multimédias IP (IMS, *IP multimedia subsystem*) est basée sur le mécanisme AKA, qui utilise un secret partagé et un protocole de type défi-réponse pour l'authentification utilisateur-réseau. Cependant, la sécurité de certains services NGN (par exemple, TVIP) est basée sur les certificats d'infrastructure de clé publique (PKI, *public key infrastructure*). Pour permettre le mélange des services NGN utilisant des certificats PKI et la sécurité IMS, il peut être souhaitable d'intégrer l'authentification basée sur l'infrastructure PKI avec l'authentification IMS en tirant parti de la sécurité du sous-système IMS.

L'intégration de l'authentification IMS avec l'authentification basée sur l'infrastructure PKI permet à l'équipement d'utilisateur et au réseau de s'authentifier mutuellement, sur la base de leurs certificats respectifs, et de s'entendre sur un ensemble de clés cryptographiques basées sur les mêmes algorithmes de génération de clé que dans le protocole AKA. A cette fin, l'équipement d'utilisateur et le réseau doivent recevoir les clés privées et certificats respectifs, et pouvoir exécuter les opérations PKI.

Pour ce qui est de l'accord concernant la clé de chiffrement (CK, *ciphering key*) et la clé d'intégrité (IK, *integrity key*), le mécanisme d'intégration décrit spécifie deux options:

1) établissement d'un accord concernant les clés *CK* et *IK* avec l'utilisation d'un secret partagé entre la fonction d'utilisateur final et l'entité S-5, à savoir l'entité fonctionnelle de profil d'utilisateur de service (SUP-FE, *service user profile functional entity*) définie dans [UIT-T Y.2012];
2) établissement d'un accord concernant les clés *CK* et *IK* sans l'utilisation d'un secret partagé.

Les flux d'appel génériques pour la première option et pour la deuxième option sont illustrés sur les Figures 5 et 6, respectivement.

6.2.6.1 Conventions

Dans le présent paragraphe, on utilise les conventions suivantes:

– "|" désigne la concaténation de chaînes.
– *CK* désigne la clé de chiffrement.
– *IK* désigne la clé d'intégrité.
– *K()* désigne un chiffrement à clé symétrique.
– N_{pr} *[]* désigne un chiffrement avec la clé privée de réseau N_{pr}.
– N_{pu} *[]*désigne un chiffrement avec la clé publique de réseau N_{pu} disponible à partir du certificat de réseau.
– U_{pr} *[]* désigne un chiffrement avec la clé privée d'utilisateur U_{pr}.

6.2.6.2 Entités intervenant dans l'authentification

- entité S-5: entité fonctionnelle de profil d'utilisateur de service (SUP-FE).
- fonction d'utilisateur final: cette entité peut prendre en charge un client SIP.
- entité S-n: entité fonctionnelle de commande de session d'appel (CSC-FE), où S-n désigne l'une des entités suivantes:
  – S-1: entité fonctionnelle serveuse de commande de session d'appel (S-CSC-FE);
  – S-2: entité fonctionnelle proxy de commande de session d'appel (P-CSC-FE);
  – S-3: entité fonctionnelle interrogatrice de commande de session d'appel (I-CSC-FE).

La notation S-n est utilisée pour désigner l'une de ces entités lorsqu'il n'y a pas de différences entre elles en ce qui concerne la procédure d'authentification décrite. Des descriptions des entités

fonctionnelles de réseau NGN (S-1, S-2, S-3, S-5 et fonction d'utilisateur final) sont données dans [UIT-T Y.2012].

6.2.6.3 Etablissement d'un accord concernant les clés CK et IK avec l'utilisation d'un secret partagé entre la fonction d'utilisateur final et l'entité S-5 (option 1)

Le flux d'appel est illustré sur la Figure 5. Les étapes de base sont les suivantes:

1) La fonction d'utilisateur final envoie à l'entité S-n la demande SIP Register contenant les identités *IMPU* et *IMPI* de l'utilisateur.

2) L'entité S-1 demande un défi aléatoire *RAND* et les clés *CK* et *IK* à l'entité S-5. Les valeurs *RAND, CK* et *IK* sont spécifiées dans [ATIS 33102].

3) L'entité S-1 reçoit les valeurs *RAND, CK* et *IK* de l'entité S-5 pour l'utilisateur.

4) L'entité S-n envoie à la fonction d'utilisateur final le message SIP 401 unauthorized contenant le défi *RAND* et sa valeur chiffrée $N_{pr}[RAND]$.

   La fonction d'utilisateur final:

   - reçoit la valeur *A*, qui est supposée égale à *RAND*, et la valeur *B*, qui est supposée égale à $N_{pr}[RAND]$;
   - récupère la clé publique de réseau N_{pu};
   - déchiffre la valeur *B* à l'aide de la clé N_{pu} et compare le résultat à la valeur *A*. Si les valeurs sont égales, le réseau est authentifié; dans le cas contraire, la procédure d'authentification est interrompue;
   - génère les clés *IK* et *CK* en utilisant le secret partagé K_s;
   - génère la valeur $U_{pr}[N_{pu}[K]|K(RAND)]$.

5) La fonction d'utilisateur final envoie à l'entité S-n le message SIP Register contenant les identités *IMPU* et *IMPI* et la valeur $U_{pr}[N_{pu}[K]|K(RAND)]$.

6) L'entité S-1 envoie à l'entité S-5 les données reçues à l'étape 5 et demande une vérification et les données de l'utilisateur.

   L'entité S-5:

   - interroge le certificat d'utilisateur pour obtenir la clé publique d'utilisateur U_{pu};
   - déchiffre à l'aide de la clé U_{pu} la valeur reçue *C*, qui est supposée égale à $U_{pr}[N_{pu}[K]|K(RAND)]$, afin de récupérer la valeur *D|E*, où *D* est supposée égale à $N_{pu}[K]$ et *E* est supposée égale à *K(RAND*);
   - déchiffre à l'aide de la clé privée de réseau N_{pr} la valeur *D* afin d'obtenir la clé *K'*;
   - déchiffre à l'aide de la clé *K'* la valeur *E* afin d'obtenir la valeur *RAND'*;
   - compare les valeurs *RAND'* et *RAND*. Si elles concordent, l'utilisateur a été authentifié.

7) L'entité S-5 communique le résultat de l'authentification et les données de l'utilisateur à l'entité S-1.

8) L'entité S-1 utilise les données pour vérifier si l'utilisateur authentifié est autorisé à s'enregistrer et à recevoir le service demandé. Si c'est le cas, l'entité S-n informe la fonction d'utilisateur final que l'accès est accordé en utilisant un message SIP 200 OK.

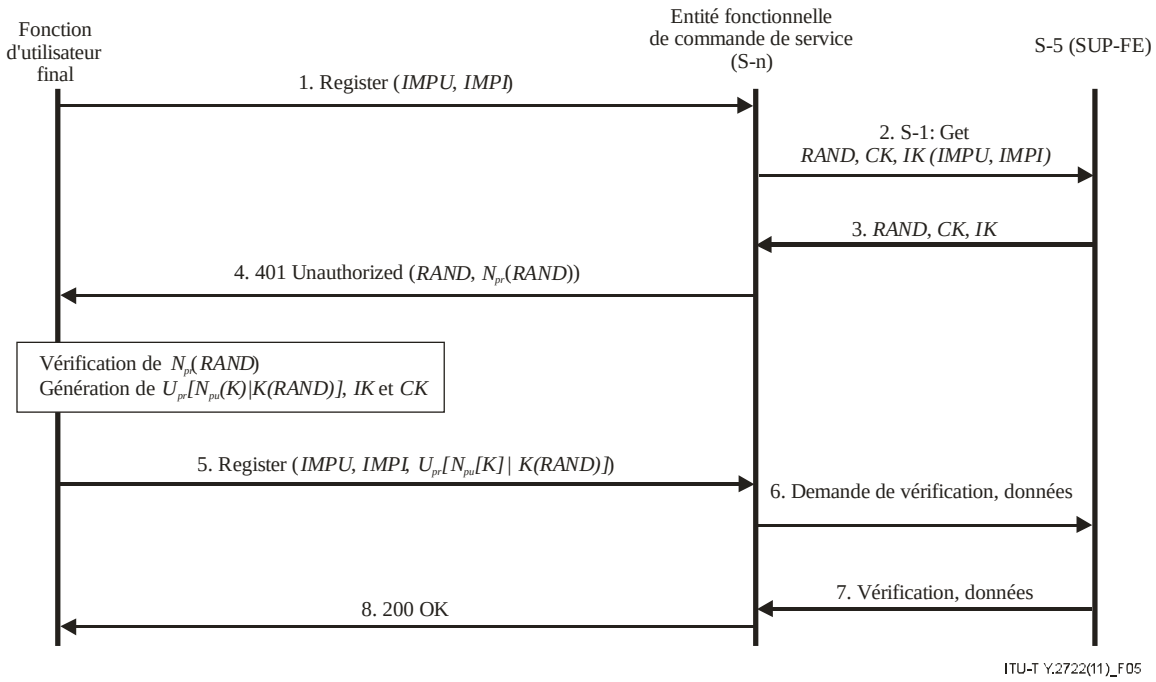


Figure 5 – Intégration du mécanisme d'authentification du sous-système IMS avec l'authentification basée sur l'infrastructure PKI (option 1)

6.2.6.4 Etablissement d'un accord concernant les clés CK et IK sans l'utilisation d'un secret partagé entre la fonction d'utilisateur final et l'entité S-5 (option 2)

Le flux d'appel est illustré sur la Figure 6. Les étapes de base sont les suivantes:

1) La fonction d'utilisateur final envoie à l'entité S-n la demande SIP Register contenant les identités *IMPU* et *IMPI* de l'utilisateur.

2) L'entité S-1 demande un défi aléatoire *RAND* à l'entité S-5. La valeur *RAND* est spécifiée dans [ATIS 33102].

3) L'entité S-1 reçoit la valeur *RAND* de l'entité S-5 pour l'utilisateur spécifié.

4) L'entité S-n envoie à la fonction d'utilisateur final le message SIP 401 unauthorized contenant le défi *RAND* et sa valeur chiffrée $N_{pr}[RAND]$.

   La fonction d'utilisateur final:

   - reçoit la valeur *A*, qui est supposée égale à *RAND*, et la valeur *B*, qui est supposée égale à $N_{pr}[RAND]$;
   - récupère la clé publique de réseau N_{pu};
   - déchiffre la valeur *B* à l'aide de la clé N_{pu} et compare le résultat à la valeur *A*. Si les valeurs sont égales, le réseau est authentifié; dans le cas contraire, la procédure d'authentification est abandonnée;
   - génère les clés *IK* et *CK* en utilisant la clé générée aléatoirement *K*;
   - génère la valeur $U_{pr}[N_{pu}[K]|K(RAND)]$.

5) La fonction d'utilisateur final envoie à l'entité S-n le message SIP Register contenant les identités *IMPU* et *IMPI* et la valeur $U_{pr}[N_{pu}[K]|K(RAND)]$.

6) L'entité S-1 envoie à l'entité S-5 les données reçues à l'étape 5 et demande une vérification, les données de l'utilisateur et les clés *CK* et *IK*.

   L'entité S-5:

   - interroge le certificat d'utilisateur pour obtenir la clé publique d'utilisateur U_{pu};

- déchiffre à l'aide de la clé U_{pu} la valeur reçue *C*, qui est supposée égale à $U_{pr}[N_{pu}[K]|K(RAND)]$, afin de récupérer la valeur *D*|*E*, où *D* est supposée égale à $N_{pu}[K]$ et *E* est supposée égale à *K(RAND*);
- déchiffre à l'aide de la clé privée de réseau N_{pr} la valeur *D* afin d'obtenir la clé *K'*;
- déchiffre à l'aide de la clé *K'* la valeur *E* afin d'obtenir la valeur *RAND'*;
- compare les valeurs *RAND'* et *RAND*. Si elles concordent, l'utilisateur a été authentifié et *K'* = *K*. Autrement dit, la clé *K* est désormais partagée entre la fonction d'utilisateur final et l'entité S-5;
- génère les clés *CK* et *IK* en utilisant la clé partagée *K*. Elle peut par exemple employer les fonctions de génération des clés *CK* et *IK* qui sont spécifiées dans [ATIS 33102] et utiliser *K* comme paramètre d'entrée.

7) L'entité S-5 communique le résultat d'authentification, les données de l'utilisateur et les clés *CK* et *IK* à l'entité S-1.

8) L'entité S-1 utilise les données pour vérifier si l'utilisateur authentifié est autorisé à s'enregistrer et à recevoir le service demandé. Si c'est le cas, l'entité S-n informe la fonction d'utilisateur final que l'accès est accordé en utilisant un message SIP 200 OK.

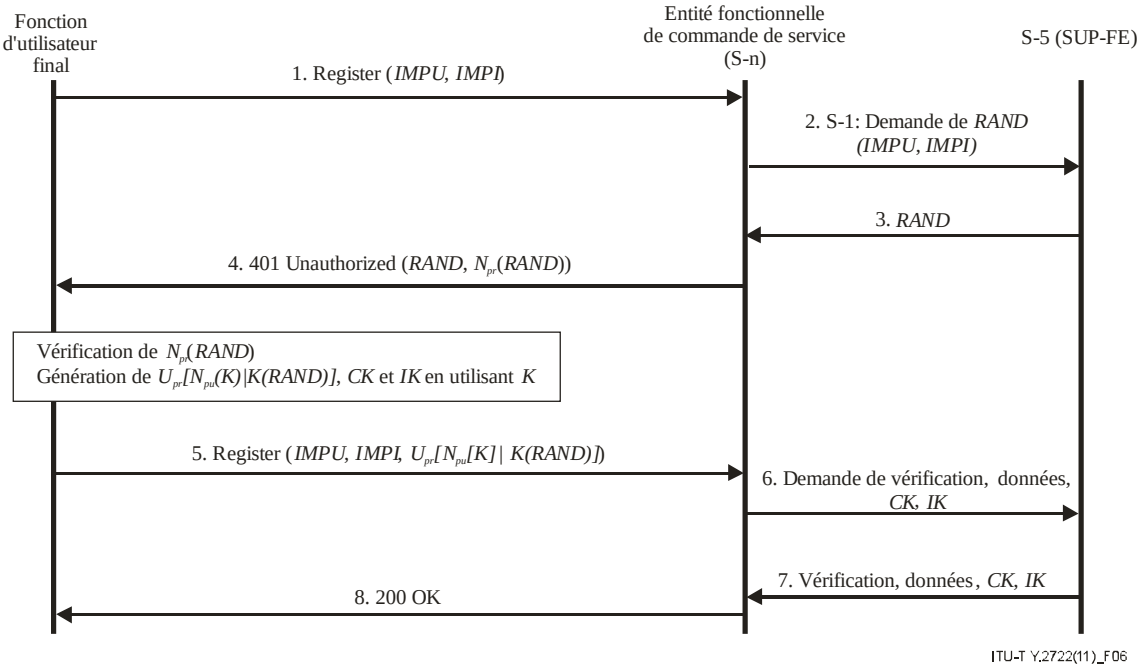


Figure 6 – Intégration du mécanisme d'authentification du sous-système IMS avec l'authentification basée sur l'infrastructure PKI (option 2)

6.2.6.5 Comparaison des options 1 et 2

Le Tableau 1 établit une comparaison des mécanismes décrits pour les options 1 et 2.

Tableau 1 – Comparaison des options 1 et 2 pour l'accord entre la fonction d'utilisateur final et l'entité S-5 concernant les clés *CK* et *IK*

	Option 1 (avec secret préalablement partagé)	**Option 2 (sans secret préalablement partagé)**
Avantages	Réutilise complètement le mécanisme AKA pour l'établissement d'un accord concernant les clés CK et IK	Ne nécessite pas la fourniture du secret partagé entre la fonction d'utilisateur final et l'entité S-5

Inconvénients	Nécessite la fourniture du secret partagé entre la fonction d'utilisateur final et l'entité S-5	Nécessite que des modifications soient apportées aux applications prises en charge par la fonction d'utilisateur final (par exemple, sur une carte à puce) et par l'entité S-5 pour qu'un accord puisse être établi concernant les clés CK et IK

Il convient de choisir l'option 1 pour simplifier l'accord concernant les clés *CK* et *IK* lorsque la fonction d'utilisateur final et l'entité S-5 partagent un secret et de choisir l'option 2 lorsque la fonction d'utilisateur final et l'entité S-5 n'ont pas de secret partagé.

Les implémentations de ce mécanisme d'intégration doivent prendre en charge les deux options.

Exigences concernant l'entité fonctionnelle S-5

Outre les capacités spécifiées dans [ATIS 33102], l'entité S-5 doit pouvoir:

- stocker les certificats des utilisateurs et du réseau dans le répertoire de certificats et les en extraire;
- exécuter le déchiffrement basé sur l'infrastructure PKI comme décrit à l'étape 6 (pour les deux options);
- appliquer le protocole Diameter modifié afin d'acheminer les informations décrites à l'étape 6 (pour les deux options) et les informations nécessaires pour la négociation avec la fonction d'utilisateur final concernant l'authentification basée sur l'infrastructure PKI;
- négocier avec la fonction d'utilisateur final un accord concernant la méthode d'authentification basée sur l'infrastructure PKI.

6.2.6.6 Exigences concernant la fonction d'utilisateur final

La fonction d'utilisateur final doit pouvoir:

- stocker en toute sécurité la clé privée d'utilisateur U_{pr};
- stocker en toute sécurité le secret partagé K_s avec le réseau (uniquement pour l'option 1);
- stocker un certificat UIT-T X.509 de réseau avec la clé publique de réseau N_{pu};
- générer aléatoirement une clé de session à usage unique *K* et réaliser un chiffrement à clé symétrique avec la clé *K*;
- générer les clés *CK* et *IK* en utilisant le secret partagé K_s comme spécifié dans [ATIS 33102] (uniquement pour l'option 1);
- générer les clés *CK* et *IK* comme décrit à l'étape 6 pour l'option 2;
- exécuter le chiffrement et le déchiffrement basés sur l'infrastructure PKI décrits aux étapes 4 et 5 pour les deux options;
- prendre en charge un client SIP avec un protocole SIP modifié pour permettre de communiquer les informations décrites aux étapes 4 et 5;
- négocier avec l'entité S-2 un accord concernant l'utilisation de l'authentification basée sur l'infrastructure PKI.

6.2.6.7 Exigences concernant l'entité S-1

Les exigences additionnelles concernant l'entité S-1 sont les suivantes:

- elle doit pouvoir élaborer des messages SIP contenant les informations décrites à l'étape 4 (pour les deux options);
- elle doit pouvoir récupérer dans les messages SIP les informations décrites à l'étape 5 et les insérer dans des messages Diameter comme décrit à l'étape 6 (pour les deux options);

- elle doit pouvoir exécuter le chiffrement basé sur l'infrastructure PKI décrit à l'étape 4 (pour les deux options);
- elle doit pouvoir comprendre la notification provenant de l'entité S-5 concernant l'utilisation de l'authentification basée sur l'infrastructure PKI.

6.2.6.8 Exigences concernant les interfaces SIP entre les entités participantes

La fonction d'utilisateur final et l'entité S-1 communiquent via les entités fonctionnelles S-2 et S-3. Les entités S-2 et S-3 n'étant pas essentielles pour l'authentification décrite, elles ne sont pas illustrées sur les Figures 5 et 6.

Des interfaces SIP sont présentes entre:

- la fonction d'utilisateur final et l'entité S-2;
- les entités S-2 et S-3;
- les entités S-1 et S-3.

A ces interfaces, il faut pouvoir négocier l'utilisation de l'authentification basée sur l'infrastructure PKI (y compris l'option spécifique pour la génération de clé) et acheminer les informations décrites aux étapes 4 et 5 (pour les deux options).

6.2.6.9 Exigences concernant les interfaces Diameter entre les entités participantes

Des interfaces Diameter sont présentes entre:

- les entités S-1 et S-5;
- les entités S-3 et S-5.

A ces interfaces, il faut pouvoir négocier l'utilisation de l'authentification basée sur l'infrastructure PKI (y compris l'option spécifique pour la génération de clé) et acheminer les informations décrites à l'étape 6 (pour les deux options).

6.2.7 Intégration de l'authentification basée sur l'infrastructure PKI avec des assertions SAML

Le langage SAML permet à une entité (par exemple, un fournisseur IdSP) d'exécuter l'authentification et à une autre entité (une partie utilisatrice, par exemple un fournisseur de service d'application) d'utiliser les résultats de l'authentification. Dans un tel scénario, un fournisseur IdSP peut implémenter de multiples méthodes d'authentification, tandis que le fournisseur de service d'application (ASP) utilise les assertions SAML du fournisseur IdSP. Ce scénario présente des avantages à la fois pour les fournisseurs IdSP et pour les fournisseurs ASP. Pour un fournisseur ASP, les avantages sont les suivants:

- le fournisseur ASP n'a pas besoin d'implémenter de nombreuses méthodes d'authentification;
- le fournisseur ASP pourrait prendre en charge une large gamme de services d'application nécessitant différentes garanties d'authentification.

Pour un fournisseur IdSP, les avantages sont les suivants:

- le fournisseur IdSP peut offrir des services IdM, en particulier l'authentification, à de multiples fournisseurs ASP;
- le fournisseur IdSP (notamment lorsqu'il s'agit d'un fournisseur NGN) peut utiliser l'infrastructure d'authentification qu'il a déployée pour offrir des services IdM à d'autres fournisseurs.

Le présent paragraphe spécifie un mécanisme permettant d'authentifier un client en utilisant des assertions SAML et une authentification basée sur l'infrastructure PKI. Ce mécanisme, ainsi que celui qui est décrit au § 6.2.6 *Intégration de l'authentification basée sur l'infrastructure PKI avec le*

sous-système IMS, permettent aux fournisseurs NGN d'utiliser leur infrastructure basée sur l'infrastructure PKI et d'en tirer parti.

Ce mécanisme est basé sur la procédure SAML *HTTP redirect binding* spécifiée dans [UIT-T X.1141].

6.2.7.1 Entités intervenant dans l'authentification et le flux d'information

- fonction d'utilisateur final: cette entité peut prendre en charge un client web et une authentification basée sur l'infrastructure PKI [UIT-T X.509];
- serveur d'application (AS): entité fournissant un service web, qui joue le rôle de partie utilisatrice et fait office de demandeur SAML comme défini dans [UIT-T X.1141];
- entité A-2: entité fonctionnelle de passerelle d'application (APL-GW-FE), qui peut exécuter l'authentification basée sur l'infrastructure PKI et faire office de répondeur SAML, comme défini dans [UIT-T X.1141];
- entité S-5: entité fonctionnelle de profil d'utilisateur de service (SUP-FE).

Le flux d'information de la procédure d'authentification est illustré sur la Figure 7. Les étapes de base de l'échange de données pour l'authentification basée sur l'infrastructure PKI avec assertion SAML sont décrites ci-dessous. Des descriptions d'entités fonctionnelles de réseau NGN (fonction d'utilisateur final, AS, A-2 et S-5) sont données dans [UIT-T Y.2012].

6.2.7.2 Conventions

Dans la description, on utilise les conventions suivantes:

"|" désigne la concaténation de chaînes;

K() désigne un chiffrement à clé symétrique;

K_s désigne un secret partagé entre l'entité A-2 et le serveur d'application;

N_{pr} *[]* désigne un chiffrement avec la clé privée de réseau N_{pr};

N_{pu} *[]* désigne un chiffrement avec la clé publique de réseau N_{pu} disponible à partir du certificat de réseau;

U_{pr} *[]* désigne un chiffrement avec la clé privée d'utilisateur U_{pr};

RAND désigne un défi généré aléatoirement.

6.2.7.3 Paramètres du mécanisme

Le présent paragraphe spécifie les paramètres propres au mécanisme. La liste des paramètres est la suivante:

`pki-auth-challenge` – Paramètre utilisé pour transmettre la valeur de *RAND*

`pki-auth-challenge-encrypted` – Paramètre utilisé pour transmettre la valeur de N_{pr}*[RAND]*

`pki-auth-user-signature` – Paramètre utilisé pour transmettre la valeur de $U_{pr}[N_{pu}[K]|K(RAND)]$

`pki-auth-keyinfo` – Paramètre utilisé pour transmettre la valeur de $K_s(K)$.

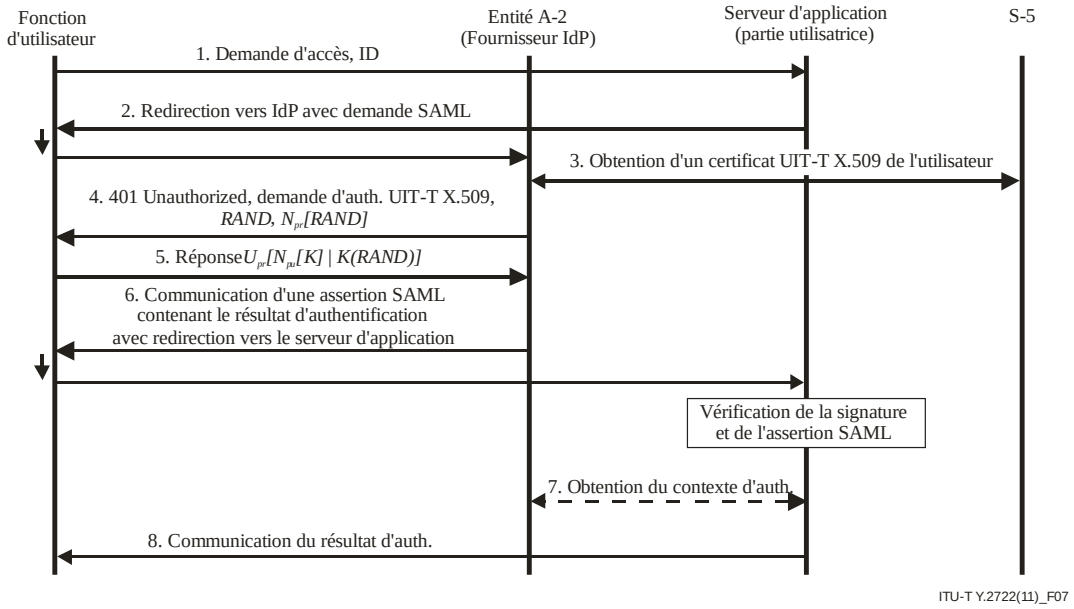


Figure 7 – Etapes de base de l'échange de données pour l'authentification basée sur l'infrastructure PKI avec assertion SAML

L'authentification mutuelle de la fonction d'utilisateur final et de l'entité A-2 est analogue à la procédure utilisée par le mécanisme d'intégration de l'authentification basée sur l'infrastructure PKI avec l'authentification IMS, qui est décrit au § 6.2.6.

Les étapes de base de la procédure d'authentification basée sur l'infrastructure PKI avec des assertions SAML sont les suivantes:

1) Un client web de la fonction d'utilisateur final envoie une demande d'accès HTTP au serveur d'application (AS). La demande contient un identificateur d'utilisateur et l'adresse URL de l'entité A-2.

2) Le serveur d'application jouant le rôle de demandeur SAML répond à la demande HTTP en envoyant une demande SAML. La demande SAML est codée dans l'en-tête de localisation de la réponse HTTP, le statut HTTP étant mis à 302 ou 303. L'agent de la fonction d'utilisateur final transmet la demande SAML en envoyant la demande HTTP GET à l'entité A-2, qui joue le rôle de répondeur SAML. Cette procédure de redirection HTTP, appelée *HTTP redirect binding*, est spécifiée dans [UIT-T X.1141]. Pour garantir l'authentification et l'intégrité du message codé URL, celui-ci doit être signé comme spécifié au § 10.2.4.5.2, *Considérations sur la sécurité* de [UIT-T X.1141]. Il convient d'utiliser le secret partagé K_s pour la signature.

3) Après validation de la signature, l'entité A-2 obtient de l'entité S-5 le certificat de l'utilisateur final et vérifie s'il est valable. Le certificat contient la clé publique de la fonction d'utilisateur final.

4) L'entité A-2 répond à la fonction d'utilisateur final par un message de réponse HTTP indiquant qu'une authentification avec utilisation d'un certificat UIT-T X.509 est nécessaire. Pour cela, la valeur de l'en-tête de réponse `WWW-Authenticate` [b-IETF RFC 2616] est mise à "`pki-auth`". Le corps du message inclut les paramètres `pki-auth-challenge` et `pki-auth-challenge-encrypted` qui acheminent respectivement la valeur du défi généré aléatoirement *RAND* et celle de son chiffrement $N_{pr}[RAND]$. L'en-tête `Content-Type` doit être mis à `application/x-www-form-urlencoded`.

5) La fonction d'utilisateur final:

   - récupère la valeur *A*, qui est supposée égale à *RAND*, et la valeur *B*, qui est supposée égale à $N_{pr}[RAND]$;

- récupère la clé publique de réseau N_{pu};
- déchiffre la valeur *B* à l'aide de la clé N_{pu} et compare le résultat à la valeur *A*. Si les valeurs sont égales, le réseau est authentifié; dans le cas contraire, la procédure d'authentification est interrompue;
- génère une clé secrète *K*;
- génère la valeur $U_{pr}[N_{pu}[K]|K(RAND)]$, met le paramètre `pki-auth-user-signature` à cette valeur, et l'envoie dans le corps d'un message HTTP POST à l'entité A-2. L'en-tête `Content-Type` du message doit être mis à la valeur `application/x-www-form-urlencoded`.

Après cette étape, l'entité A-2 vérifie si la réponse est valable. A cette fin, elle:

- interroge le certificat d'utilisateur pour obtenir la clé publique d'utilisateur U_{pu};
- déchiffre à l'aide de la clé U_{pu} la valeur reçue *C*, qui est supposée égale à $U_{pr}[N_{pu}[K]|K(RAND)]$, afin de récupérer la valeur *D*|*E*, où *D* est supposée égale à $N_{pu}[K]$ et *E* est supposée égale à *K(RAND*);
- déchiffre à l'aide de la clé privée de réseau N_{pr} la valeur *D* afin d'obtenir la clé *K'*;
- déchiffre à l'aide de la clé *K'* la valeur *E* afin d'obtenir la valeur *RAND'*;
- compare les valeurs *RAND'* et *RAND*. Si elles concordent, l'utilisateur a été authentifié et *K'* = *K*. Autrement dit, la clé *K* est désormais partagée entre la fonction d'utilisateur final et l'entité A-2.

6) Si toutes les étapes ci-dessus ont abouti, l'entité A-2:

- génère une assertion SAML dans laquelle l'attribut de méthode de l'élément <SubjectConfirmation> est mis à la valeur sender-vouches;
- calcule la valeur $K_s(K)$;
- inclut l'assertion dans une réponse SAML. Elle transmet ensuite la réponse SAML et la valeur calculée $K_s(K)$ sur HTTP comme décrit pour la demande SAML à l'étape 2 (c'est-à-dire dans le cadre d'une chaîne de requête). La valeur $K_s(K)$ est acheminée par le paramètre `pki-auth-keyinfo`;
- pour garantir l'authentification de l'origine et l'intégrité du message codé URL, l'entité A-2 le signe comme spécifié au § 10.2.4.5.2, *Considérations sur la sécurité* de [UIT-T X.1141]. Il convient d'utiliser le secret partagé K_s pour la signature.

  Après validation de l'URL signée, le serveur d'application est assuré que l'assertion SAML est faite par l'entité A-2. Le serveur d'application vérifie l'assertion proprement dite (par exemple pour garantir que les conditions sont respectées). Après cela, il récupère la valeur $K_s(K)$ et la déchiffre à l'aide du secret partagé K_s afin d'obtenir la clé *K*. A ce stade, le serveur d'application a authentifié la fonction d'utilisateur final et la clé *K* est partagée entre les deux entités, qui peuvent l'utiliser pour sécuriser les communications entre elles.

7) Si la politique l'exige aux fins de la prise d'une décision d'autorisation, le serveur d'application obtient des informations au sujet du contexte d'authentification. Dans ce cas, l'entité A-2 répond avec des informations spécifiées par la classe de contexte d'authentification de clé publique – UIT-T X.509 [UIT-T X.1141].

8) Le serveur d'application envoie à la fonction d'utilisateur final le résultat de la décision d'autorisation.

6.2.7.4 Exigences additionnelles concernant les entités participant à l'authentification

Afin de prendre en charge le mécanisme décrit, les entités participantes doivent satisfaire aux exigences suivantes:

6.2.7.4.1 Exigences concernant la fonction d'utilisateur final

La fonction d'utilisateur final doit pouvoir:

- prendre en charge un client HTTP;
- stocker en toute sécurité sa clé privée U_{pr} (par exemple sur une carte à puce);
- obtenir la clé publique de réseau N_{pu};
- exécuter un chiffrement et un déchiffrement;
- générer une clé K.

6.2.7.4.2 Exigences concernant le serveur d'application (AS)

- le serveur d'application doit prendre en charge le langage SAML [UIT-T X.1141];
- le serveur d'application doit avoir un secret partagé (K_s) avec l'entité A-2.

6.2.7.4.3 Exigences concernant l'entité fonctionnelle A-2

L'entité fonctionnelle A-2 doit pouvoir:

- prendre en charge le protocole HTTP;
- stocker en toute sécurité sa clé privée N_{pr};
- obtenir la clé publique d'utilisateur U_{pu};
- exécuter un chiffrement et un déchiffrement;
- générer un défi aléatoire RAND;
- prendre en charge le langage SAML [UIT-T X.1141];
- avoir un secret partagé (K_s) avec le serveur d'application.

6.2.7.4.4 Exigences concernant l'entité fonctionnelle S-5

L'entité fonctionnelle S-5 devrait être capable de stocker les certificats UIT-T X.509 des utilisateurs ou de les extraire du répertoire (ou d'effectuer ces deux actions).

6.2.7.5 Exigences additionnelles concernant les interfaces entre les entités participantes

Les exigences concernant les interfaces sont les suivantes:

- l'interface entre la fonction d'utilisateur final et le serveur d'application doit prendre en charge le protocole HTTP [b-IETF RFC 2616];
- les interfaces entre la fonction d'utilisateur final et les entités fonctionnelles A-2 doivent prendre en charge le protocole HTTP [b-IETF RFC 2616];
- l'interface entre l'entité A-2 et le serveur d'application doit prendre en charge le langage SAML [UIT-T X.1141];
- l'interface entre les entités fonctionnelles A-2 et S-5 doit prendre en charge un mécanisme de requête-réponse qui permet à l'entité A-2 d'obtenir les certificats X.509 des utilisateurs auprès de l'entité S-5.

6.2.8 Intégration de l'authentification basée sur OpenID avec l'authentification AKA

L'intégration de l'authentification AKA avec l'authentification basée sur OpenID permet une combinaison des capacités IdM centrées sur le réseau et des capacités IdM centrées sur l'utilisateur. Le mécanisme d'intégration:

- permet aux fournisseurs de réseau de fournir des services d'identité aux utilisateurs accédant aux applications web;

- peut être utilisé pour offrir aux utilisateurs une authentification unique à travers le réseau IMS et l'environnement des services web à l'aide d'une application ISIM existante et d'autres applications SIM qui reposent sur AKA;
- permet aux utilisateurs d'avoir le contrôle de leurs identificateurs publics sur le web comme spécifié dans [b-OpenID v.2] tout en tirant parti des services NGN;
- améliore la sécurité des utilisateurs en employant un fournisseur de réseau fiable pour les utilisateurs en ce qui concerne le contrôle d'accès aux applications web.

[b-3GPP TR 33.924] décrit plusieurs solutions relatives à l'intégration d'OpenID avec AKA reposant sur l'utilisation d'une fonction de serveur d'amorçage (BSF, *bootstrapping server function*).

Le présent paragraphe décrit un mécanisme additionnel pour l'intégration d'OpenID et d'AKA. A cette fin, la spécification OpenID fait appel à divers mécanismes d'authentification.

OpenID peut interfonctionner avec d'autres technologies, par exemple OAuth, comme indiqué dans l'Appendice II.

6.2.8.1 Entités intervenant dans l'authentification et le flux d'information

- Fonction d'utilisateur final: cette entité peut prendre en charge un client web et communiquer avec l'application SIM appropriée;
- serveur d'application: entité fournissant un service web, qui joue le rôle de partie utilisatrice;
- entité A-2: entité fonctionnelle de passerelle d'application (APL-GW-FE), qui peut jouer le rôle de fournisseur d'identité OpenID [b-OpenID v.2]. (L'entité A-2 partage, à titre d'option, un secret à court terme avec le serveur d'application comme spécifié dans [b-OpenID v.2]);
- entité S-5: entité fonctionnelle de profil d'utilisateur de service (SUP-FE).

Le flux d'information de la procédure d'authentification est illustré sur la Figure 8. La procédure d'établissement de la clé de signature à court terme entre le serveur d'application et l'entité A-2 n'est pas représentée. La figure illustre les étapes de base de la procédure pour deux options OpenID:

a) l'entité A-2 et le serveur d'application partagent un secret;

b) l'entité A-2 et le serveur d'application ne partagent pas de secret.

Les étapes 1 à 6 sont communes aux deux options. L'étape 7a concerne uniquement l'option a).

Les étapes 7b, 8b et 9b concernent uniquement l'option b).

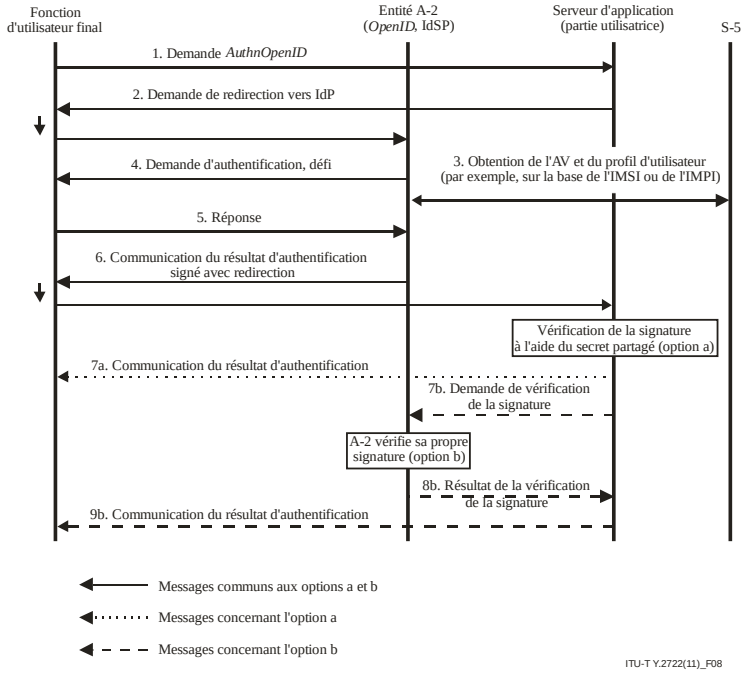


Figure 8 – Intégration du mécanisme d'authentification AKA avec OpenID

Les étapes de base sont les suivantes:

1) Un client web de la fonction d'utilisateur final envoie une demande d'authentification *AuthnOpenID* au serveur d'application. La demande inclut un identificateur OpenID.

2) Le serveur d'application utilise l'identificateur OpenID présenté pour découvrir l'adresse URL de l'entité A-2, qui joue le rôle de fournisseur d'identité OpenID, et redirige la demande d'authentification d'utilisateur vers cette adresse URL.

   Après cette étape, l'entité A-2 corrèle l'identificateur de l'utilisateur avec l'identité appropriée (telle qu'IMSI ou IMPI).

3) L'entité A-2 obtient de l'entité S-5 le vecteur d'authentification (AV) AKA et le profil d'utilisateur sur la base de l'identité IMPI.

4) L'entité A-2 envoie à la fonction d'utilisateur final la demande d'authentification en utilisant la méthode HTTP Digest AKA [b-IETF RFC 4169] ou [b-IETF RFC 3310]. La demande inclut un défi et une grandeur qui permet à la fonction d'utilisateur final d'authentifier le réseau.

   Après cette étape, la fonction d'utilisateur final authentifie le réseau, comme spécifié dans [b-IETF RFC 4169] ou [b-IETF RFC 3310].

5) La fonction d'utilisateur final envoie la réponse au défi à l'entité A-2, comme spécifié dans [b-IETF RFC 4169] ou [b-IETF RFC 3310].

   Après cette étape, l'entité A-2 authentifie la fonction d'utilisateur final comme spécifié dans [b-IETF RFC 4169] ou [b-IETF RFC 3310].

6) L'entité A-2 envoie à la fonction d'utilisateur final un message signé assertant que l'identificateur OpenID déclaré appartient à l'utilisateur. Le message est signé à l'aide d'un secret partagé avec le serveur d'application pour l'option a). Pour l'option b), le message est signé avec la clé secrète de l'entité A-2. Le message inclut une demande de redirection du client web de la fonction d'utilisateur final vers le serveur d'application. Les procédures de signature et de redirection sont décrites en détail dans [b-OpenID v.2]. [b-OpenID v.2] spécifie également des mesures de prévention des attaques basées sur la réutilisation de l'assertion d'authentification signée.

Etapes qui sont propres à l'option a):

7a) Après avoir vérifié la signature de la réponse reçue à l'étape 6, le serveur d'application communique à la fonction d'utilisateur final le résultat de l'authentification. Le serveur d'application utilise le secret partagé avec l'entité A-2 pour cette vérification.

En cas d'échec de l'une des étapes 1 à 6 ou de l'étape 7a, la procédure d'authentification s'arrête.

Etapes qui sont propres à l'option b):

7b) Le serveur d'application envoie à l'entité A-2 une copie du message reçu à l'étape 6 avec une demande de vérification de la signature.

8b) Après avoir vérifié sa propre signature, l'entité A-2 communique le résultat de la vérification au serveur d'application.

9b) Le serveur d'application communique le résultat de l'authentification à la fonction d'utilisateur final.

En cas d'échec de l'une des étapes 1 à 6 ou de l'étape 7b, 8b ou 9b, la procédure d'authentification s'arrête.

6.2.8.2 Exigences additionnelles concernant les entités participant à l'authentification

Afin de prendre en charge le mécanisme décrit, les entités participantes doivent satisfaire aux exigences suivantes:

6.2.8.2.1 Exigences concernant la fonction d'utilisateur final

La fonction d'utilisateur final doit pouvoir:

- procéder à une authentification à l'aide de la méthode HTTP Digest AKA;
- communiquer avec l'application SIM appropriée.

6.2.8.2.2 Exigences concernant le serveur d'application

Le serveur d'application doit pouvoir prendre en charge la spécification OpenID version 2.0 [b-OpenID v.2].

6.2.8.2.3 Exigences concernant l'entité fonctionnelle A-2

L'entité fonctionnelle A-2 doit pouvoir:

- procéder à une authentification HTTP Digest AKA;
- corréler l'identificateur OpenID de l'utilisateur avec l'identificateur approprié (par exemple IMSI ou IMPI);
- jouer le rôle de fournisseur d'identité OpenID.

6.2.8.2.4 Exigences concernant l'entité fonctionnelle S-5

Il n'y a pas d'exigences concernant l'entité fonctionnelle S-5 autres que celles spécifiées dans [UIT-T Y.2012].

6.2.8.3 Exigences additionnelles concernant les interfaces entre les entités participantes

Les exigences concernant les interfaces sont les suivantes:

- l'interface entre la fonction d'utilisateur final et le serveur d'application doit prendre en charge l'authentification OpenID comme spécifié dans [b-OpenID v.2];
- les interfaces entre la fonction d'utilisateur final et les entités fonctionnelles A-2 doivent prendre en charge le protocole HTTP Digest AKA [b-IETF RFC 4169] ou [b-IETF RFC 3310].

Concernant l'interface entre les entités fonctionnelles A-2 et S-5, il n'y a pas d'exigences propres au mécanisme.

6.2.8.4 Mécanisme d'interfonctionnement OpenID et AKA pour le scénario de terminal d'utilisateur partagé

Le mécanisme décrit dans le présent paragraphe prend également en charge le scénario de terminal partagé, décrit dans [b-3GPP TR 33.924]. Le scénario de terminal d'utilisateur partagé désigne une situation dans laquelle un agent d'authentification (une entité ayant accès à la carte UICC) et l'agent de navigation ne sont pas situés sur le même terminal d'utilisateur.

Si on considère que dans la solution AKA directe spécifiée dans le présent paragraphe, le fournisseur IdSP correspond au regroupement des fonctions NAF/BSF, les scénarios décrits dans [b-3GPP TR 33.924] sont complètement pris en charge par la solution. Le mécanisme repose sur l'authentification AKA directe et non sur l'authentification basée sur l'architecture GBA.

6.2.9 Architecture d'amorçage générique (GBA)

L'architecture d'amorçage générique (GBA, *generic bootstrapping architecture*) spécifie un cadre pour amorcer l'authentification et établir un accord de clé sur la base du mécanisme 3GPP d'authentification et d'accord de clé (AKA). L'architecture GBA facilite l'authentification des utilisateurs finals auprès de la fonction d'application de réseau (NAF, *network application function*) et peut être utilisée dans le cadre de la gestion d'identité dans les réseaux NGN pour permettre:

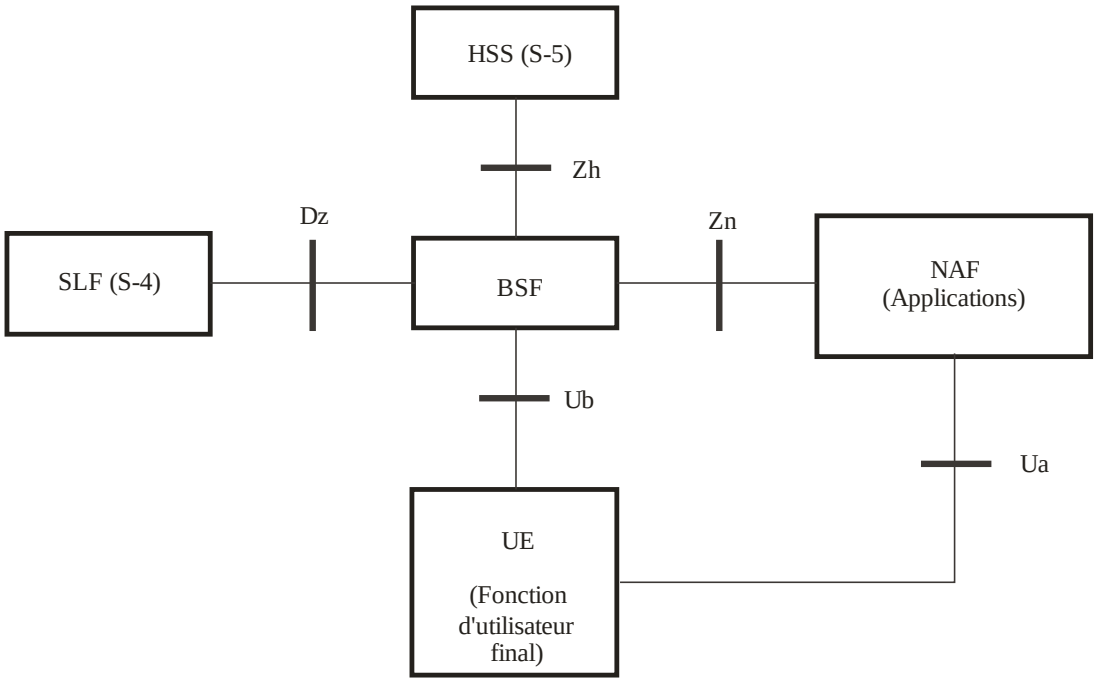
- l'authentification et l'accord de clé;
- la protection de la confidentialité;
- l'authentification unique.

L'architecture GBA est un système d'authentification qui comporte trois parties:

- un utilisateur final, qui essaie d'obtenir des services de réseau à partir d'un équipement d'utilisateur (UE);
- un serveur d'application (appelé fonction d'application de réseau (NAF));
- une entité de confiance (appelée fonction de serveur d'amorçage (BSF)), qui participe à l'authentification et à l'échange de clé entre les deux autres entités.

L'architecture GBA permet l'authentification de l'utilisateur final, qui utilise l'équipement d'utilisateur, auprès d'un serveur d'application (NAF) sans révéler à la fonction NAF les secrets et justificatifs à long terme de l'utilisateur final grâce à l'utilisation d'une entité de confiance (BSF).

Dans la Figure 9, on a représenté l'architecture GBA [b-ETSI TS 133 220] et indiqué la correspondance entre les entités définies par 3GPP et les entités fonctionnelles spécifiées dans [UIT-T Y.2012].



NOTE – Les entités entre parenthèses sont les entités spécifiées dans [ITU-T Y.2012].

ITU-T Y.2722(11)_F09

Figure 9 – Modèle simple de réseau pour l'amorçage

Les étapes de base de la procédure GBA sont les suivantes:

1) La fonction NAF demande l'authentification et négocie l'utilisation de l'architecture GBA via le point de référence Ua.

2) Le client BSF pris en charge par l'équipement d'utilisateur lance la procédure d'amorçage via le point de référence Ub. La fonction BSF obtient les informations d'authentification et les paramètres de sécurité d'utilisateur GBA auprès du serveur HSS via le point de référence Zh. L'équipement d'utilisateur et la fonction BSF s'authentifient mutuellement en utilisant la procédure HTTP Digest AKA. L'équipement d'utilisateur reçoit alors de la fonction BSF un identificateur de transaction d'amorçage (B-TID) et établit une clé partagée (Ks) avec la fonction BSF.

3) L'équipement d'utilisateur déduit la clé Ks_NAF de la clé Ks et envoie l'identificateur B-TID (ainsi que les données propres à l'application) à la fonction NAF.

4) La fonction NAF envoie l'identificateur B-TID à la fonction BSF via le point de référence Zn.

5) Compte tenu de l'identificateur B-TID, la fonction BSF détermine la clé Ks qui doit être utilisée, en déduit la clé Ks_NAF et envoie cette dernière à la fonction NAF.

6) Enfin, l'équipement d'utilisateur et la fonction NAF peuvent s'authentifier mutuellement en utilisant la clé partagée Ks_NAF. La procédure exacte d'authentification dépend du protocole utilisé entre l'équipement d'utilisateur et la fonction NAF. Par exemple, selon les spécifications de l'architecture GBA, les applications basées sur le protocole HTTP peuvent utiliser soit la procédure d'authentification HTTP Digest [b-IETF RFC 2617] soit les suites de chiffrement à clé préalablement partagée TLS [b-IETF RFC 4279].

NOTE – La fonction BSF interroge la fonction SLF via le point de référence Dz pour obtenir le nom du serveur HSS contenant les données propres à l'abonné. La fonction SLF n'est pas nécessaire lorsque la fonction BSF est configurée pour utiliser un serveur HSS prédéfini.

La correspondance entre les entités GBA et les entités NGN spécifiées dans [UIT-T Y.2012], *Prescriptions et architecture fonctionnelles des réseaux de prochaine génération*, est la suivante:

- la fonction NAF correspond à l'entité "Applications" illustrée sur la Figure 3, *Architecture fonctionnelle généralisée du réseau NGN*, de [UIT-T Y.2012];
- le serveur HSS correspond à l'entité fonctionnelle S-5 de profil d'utilisateur de service;
- la fonction SLF correspond à l'entité fonctionnelle S-4 de localisation d'abonnement;
- l'équipement d'utilisateur correspond à la fonction d'utilisateur final.

6.2.10 Authentification basée sur l'identité IMSI

Suivant le niveau de garantie requis, l'identité internationale d'abonné mobile (IMSI, *international mobile subscriber identity*) peut être utilisée dans le réseau de protocole d'application hertzienne (WAP, *wireless application protocol*) pour l'authentification afin d'assurer la rétrocompatibilité. Etant donné que l'IMSI est une chaîne alphabétique unique, elle peut être utilisée comme l'identité d'une entité pour un service particulier.

L'approche:

- utilise le code IMSI comme identité d'entité dans les applications sans fil;
- offre un canal de service fiable à condition que le point d'extrémité ait une identité IMSI légitime lorsqu'il demande une authentification;
- repose sur l'hypothèse selon laquelle tous les systèmes ont confiance dans le résultat d'authentification de la passerelle WAP et offrent des services pour cette entité;
- peut être utilisée pour assurer une fonction d'authentification unique en vertu de l'unicité de l'identité IMSI pour le même point d'extrémité entre le terminal GPRS/CDMA 1x et les applications sans fil (par exemple boîte de messagerie de terminal sans fil, etc.);
- assure la sécurité du code IMSI.

6.3 Corrélation et lien

D'après [UIT-T Y.2720], *Cadre de gestion d'identité des réseaux NGN, les informations d'identité (par exemple, les identificateurs, les justificatifs d'identité et les attributs) peuvent faire l'objet d'une corrélation afin d'établir un lien pour garantir l'identité d'une entité.*

Une solution de corrélation a notamment pour objectif de rassembler divers types d'informations d'identité provenant de différentes sources et de les présenter aux applications dans un format unifié qu'elles sont à même de comprendre.

Le concept d'une telle solution est illustré sur la Figure 10, sur laquelle sont représentés trois exemples de source d'informations d'identité: un serveur HSS, un serveur de présence et une base de données contenant les données d'utilisateur propres à l'application. Une application peut avoir besoin des trois types d'informations pour prendre des décisions d'authentification et d'autorisation. Dans l'exemple illustré, le mécanisme de corrélation emploie les protocoles Diameter, LDAP et SQL pour obtenir les données auprès des différentes sources. Ces données sont ensuite présentées à l'application dans un format qu'elle comprend. Le mécanisme de corrélation évite donc aux applications de devoir prendre en charge plusieurs protocoles pour communiquer avec les différentes sources d'une information d'identité.

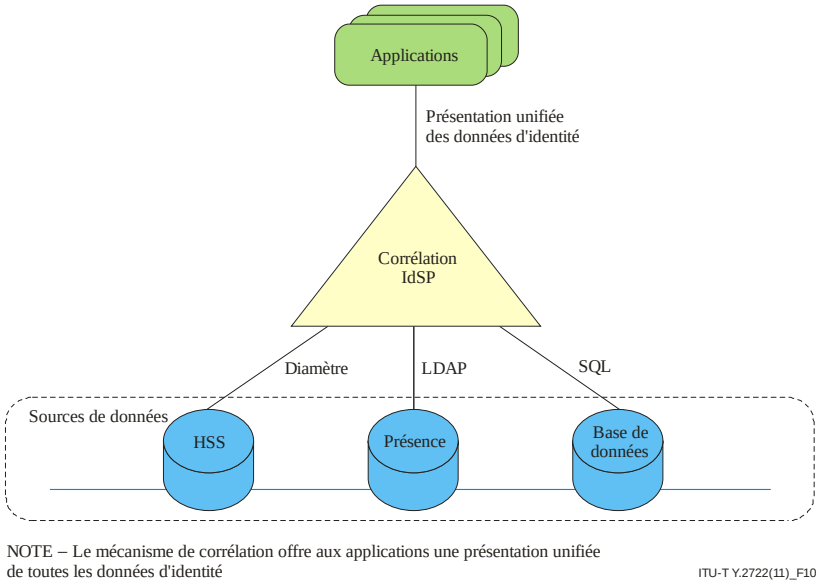


Figure 10 – Corrélation des informations d'identité

6.4 Découverte

D'après [UIT-T Y.2721], *Spécifications et cas d'utilisation de la gestion d'identité dans les réseaux NGN, il est obligatoire que le fournisseur IdSP/NGN prenne en charge des fonctions et des capacités permettant de découvrir les sources d'informations d'identité* dans son domaine et à travers les domaines de différents fournisseurs IdSP/NGN.

Le présent paragraphe donne des exemples de mécanismes standards qui prennent en charge ces exigences et les références aux spécifications pertinentes.

6.4.1 Découverte intraréseau

[UIT-T Y.2012], *Prescriptions et architecture fonctionnelles du réseau de prochaine génération*, définit une entité spéciale – l'entité fonctionnelle de localisation d'abonnement (SL-FE) – qui a l'adresse de l'entité fonctionnelle de profil d'utilisateur de service (SUP-FE) qui stocke les informations d'identité d'un abonné particulier. L'entité SL-FE permet de découvrir l'entité SUP-FE qui est chargée de stocker les profils d'utilisateur, les données de localisation relatives à l'abonné et les données relatives au statut de présence. En interrogeant l'entité SUP-FE, les entités de réseau peuvent obtenir ces informations d'identité. Comme spécifié dans [UIT-T Y.2012], les entités de réseau suivantes peuvent interroger l'entité SL-FE pour obtenir l'adresse de l'entité SUP-FE appropriée:

- entité fonctionnelle de prise en charge d'application (AS-FE);
- entité fonctionnelle interrogatrice de commande de session d'appel (I-CSC-FE);
- entité fonctionnelle serveuse de commande de session d'appel (S-CSC-FE).

Un mécanisme permettant à ces entités de trouver dans le réseau d'un opérateur l'adresse de l'entité SUP-FE qui stocke les informations d'identité d'un utilisateur donné est spécifié dans [3GPP TS 23.228]. Il est à noter que la correspondance entre les entités de [UIT-T Y.2012] et les entités de [3GPP TS 23.228] est la suivante:

- AS-FE correspond à AS;
- I-CSC-FE correspond à I-CSCF;
- S-CSC-FE correspond à S-CSCF;
- SL-FE correspond à SLF.

6.4.2 Découverte interréseaux

Parmi les exemples de mécanismes de découverte interréseaux d'un fournisseur IdSP figurent ceux qui sont décrits dans les spécifications SAML [UIT-T X.1141] et ID-WSF [b-LA WSF]. Ces mécanismes dépendent d'accords préétablis entre les entités concernées (par exemple un fournisseur IdSP et une partie utilisatrice) ou les membres d'une fédération.

Un autre exemple est celui de OpenID [b-OpenID v.2] qui spécifie un mécanisme de découverte qui permet à une partie utilisatrice de localiser le fournisseur IdSP d'un utilisateur à l'aide de l'identificateur Open ID fourni par l'utilisateur.

6.5 Communications et échanges d'informations IdM

Le présent paragraphe recommande des protocoles et des mécanismes applicables aux communications et aux échanges d'informations d'identité.

6.5.1 Sécurité des communications et échanges IdM

Le présent paragraphe recommande des mécanismes permettant de protéger l'intégrité et la confidentialité des communications IdM.

6.5.1.1 Solutions basées sur SAML 2.0 [UIT-T X.1141]

Concernant la protection de l'intégrité et de la confidentialité, la spécification SAML 2.0 recommande d'utiliser un canal sécurisé ou un protocole de réseau sécurisé comme TLS ou IPSec et de le configurer de manière à protéger les paquets transmis via la connexion de réseau.

Concernant la protection de l'intégrité au niveau des messages, on peut utiliser, en plus du canal de communication sécurisé, une signature XML. Il faut suivre le § 8.4, "SAML et syntaxe et traitement de signature XML", de [UIT-T X.1141] lorsqu'une signature XML est utilisée.

Concernant la protection de la confidentialité au niveau des messages, on peut utiliser, en plus du canal de communication sécurisé, un chiffrement XML. Il faut suivre le § 8.4, "SAML et syntaxe et traitement de signature XML", de [UIT-T X.1141] lorsqu'un chiffrement XML est utilisé.

6.5.1.2 Cadre des services web d'identité (ou ID-WSF)

Afin d'utiliser le cadre ID-WSF, l'intégrité et la confidentialité des communications et des messages entre l'expéditeur et le destinataire devraient être protégées. Comme dans le cas de la spécification SAML 2.0, il est recommandé d'utiliser un canal sécurisé ou un protocole de réseau sécurisé comme TLS ou IPSec et de le configurer de manière à protéger les paquets transmis via la connexion de réseau [b-LA ID-WSF security].

1) Protection de canal dans la couche transport

En cas d'utilisation de SSL ou de TLS comme protocole de réseau sécurisé pour le cadre ID-WSF, il faut utiliser la version SSL 3.0 ou la version TLS 1.0 ou une version supérieure. L'entité de destination d'une connexion SSL (3.0) ou TLS (1.0) doit offrir ou accepter des suites de chiffrement appropriées pendant la prise de contact. Les suites de chiffrement TLS 1.0 (ou les suites SSL 3.0 équivalentes) recommandées sont les suivantes (la liste n'est pas exhaustive):

- TLS_RSA_WITH_RC4_128_SHA
- TLS_RSA_WITH_3DES_EDE_CBC_SHA
- TLS_DHE_DSS_WITH_3DES_EDE_CBC_SHA
- TLS_RSA_WITH_AES_CBC_SHA
- TLS_DHE_DSS_WITH_AES_CBC_SHA

Pour la signature et la vérification des messages de protocole, il est recommandé aux entités en communication d'utiliser des certificats et des clés privées qui sont distincts des certificats et des clés privées appliquées pour la protection de canal SSL ou TLS.

On peut utiliser d'autres protocoles de sécurité comme IPSec ou Kerberos sous réserve qu'ils implémentent des mesures de sécurité équivalentes.

2) Protection de la confidentialité des messages

En présence d'intermédiaires, les entités en communication sont tenues de faire en sorte que les informations sensibles ne soient pas divulguées à des entités non autorisées. Dans ce cas, ces entités sont tenues d'utiliser les mécanismes de confidentialité spécifiés dans [b-OASIS WSS SOAP] pour chiffrer le contenu <S:Body> de l'enveloppe SOAP.

3) Règles concernant l'intégrité des messages

Dans le présent paragraphe, les règles concernant l'intégrité des messages ne s'appliquent que si on utilise [b-OASIS WSS SOAP] pour un message de protocole ID-WSF lié à SOAP conformément à [b-LA SOAP binding].

Dans ce cas, l'expéditeur doit créer un seul élément <ds:Signature> contenu dans l'en-tête <wsse:Security> et cette signature doit faire référence à tous les composants de message qui doivent être signés.

En particulier, cette signature doit faire référence à l'élément SOAP Body (l'élément proprement dit), le jeton de sécurité associé à la signature et tous les en-têtes du message qui ont été définis dans [b-LA SOAP binding], y compris les blocs d'en-tête obligatoires et les blocs d'en-tête facultatifs.

L'élément <saml2:Assertion> acheminé dans l'en-tête <wsse:Security> est un exemple de jeton de sécurité. L'en-tête wsu:Timestamp du bloc d'en-tête wsse:Security et les blocs d'en-tête wsa:MessageID, wsa:RelatesTo, sb:Framework, sb:Sender et sb:InvocationIdentity sont des exemples d'éléments d'en-tête auxquels une signature doit en principe faire référence.

Il est à noter qu'il convient de procéder avec soin à l'élaboration des éléments contenus dans les paramètres de référence figurant dans les références de point d'extrémité, étant donné qu'ils deviendront des blocs d'en-tête SOAP. Des mesures appropriées devraient être prises pour éviter d'avoir des attributs d'identification incompatibles ou en double, par exemple, en utilisant des techniques permettant de générer des identificateurs uniques.

Si le message est signé, l'expéditeur est tenu d'inclure la signature XML résultante dans un élément <ds:Signature> contenu dans l'en-tête <wsse:Security>.

L'élément <ds:Signature> doit faire référence à la clé de confirmation du sujet avec un élément <ds:KeyInfo>. L'élément <ds:KeyInfo> doit inclure un élément <wsse:SecurityTokenReference> de manière à pouvoir localiser la clé de confirmation du sujet dans l'en-tête <wsse:Security>. Il est recommandé que l'inclusion de la référence se fasse conformément aux indications données au § 3.4.2 de [b-OASIS SAML token].

i) Règles de traitement au niveau de l'expéditeur

- L'élaboration de l'élément d'en-tête <wsse:Security> doit se faire conformément aux règles spécifiées dans [b-OASIS SAML token].
- L'élément d'en-tête <wsse:Security> doit avoir un attribut mustUnderstand avec la valeur logique *vrai*.
- L'expéditeur est tenu de faire figurer le jeton de sécurité d'authentification de message directement dans l'élément <wsse:Security>.
- L'expéditeur est tenu de suivre les règles d'intégrité des messages énoncées pour les expéditeurs et les destinataires lorsque des mécanismes d'authentification de message sont utilisés.

Les considérations suivantes ne s'appliquent pas aux jetons de support:

- Dans le cas où les paramètres de déploiement nécessitent une authentification de message indépendante, pour accomplir l'obligation, il faut signer le corps du message et des parties de l'en-tête et faire figurer l'élément <ds:Signature> directement dans l'en-tête <wsse:Security>.
- Dans le cas où les paramètres de déploiement ne nécessitent pas une authentification de message indépendante, pour accomplir l'obligation de confirmation du sujet, on peut corréler le certificat et la clé utilisés pour l'authentification d'entité homologue avec le certificat et la clé décrits par le jeton d'authentification de message. Pour cela, l'autorité assertante doit élaborer l'assertion de manière à pouvoir vérifier sans ambiguïté que la clé de confirmation correspond bien au certificat et à la clé utilisés pour établir l'authentification de l'entité homologue. Cela est nécessaire pour réduire la menace d'une attaque visant à remplacer un certificat. Il est recommandé que le certificat ou la chaîne de certificats soit lié à la clé de confirmation du sujet.

ii) Règles de traitement au niveau du destinataire

- Le destinataire est tenu de localiser l'élément <wsse:Security> qui lui est destiné. Cela doit se faire conformément aux règles spécifiées dans [b-OASIS WSS SOAP] et aux profils de jeton WSS applicables (par exemple [b-OASIS SAML token] pour les jetons SAML).
- L'élément d'en-tête <wsse:Security> doit avoir un attribut mustUnderstand avec la valeur logique *vrai* et le destinataire doit pouvoir traiter ce bloc d'en-tête conformément à [b-OASIS WSS SOAP] et aux profils de jeton WSS appropriés (par exemple [b-OASIS SAML token] pour les jetons SAML).
- Le destinataire est tenu de localiser le jeton de sécurité et de déterminer que l'autorité qui a délivré le jeton est fiable.
- Le destinataire est tenu de valider la signature de l'émetteur pour le jeton. Cette validation doit se faire conformément aux règles de validation essentielles décrites dans [b-W3C XML signature]. Il est recommandé que le destinataire valide la fiabilité de la sémantique de la clé de signature, en fonction du risque que l'authentification soit incorrecte.
- Si le message a été signé, le destinataire est tenu de localiser l'élément <ds:Signature> acheminé dans l'en-tête <wsse:Security>.
- Lorsque le mécanisme de sécurité n'est pas peerSAMLV2, le destinataire est tenu de déterminer le contenu de l'élément <ds:KeyInfo> acheminé dans l'élément <ds:Signature> et d'utiliser la clé décrite pour valider les éléments signés. Lorsque le mécanisme de sécurité est peerSAMLV2, la clé est la clé de client utilisée dans l'authentification de client SSL/TLS.
- Le destinataire est tenu de suivre les règles d'intégrité des messages énoncées pour les expéditeurs et les destinataires lorsque des mécanismes d'authentification de message sont utilisés.

4) Traitement des messages avec un jeton WSS UIT-T X.509

La sémantique et les règles de traitement applicables aux messages UIT-T X.509 sont décrites dans le présent paragraphe. On trouvera un exemple dans l'Appendice I.

Les URI prenant en charge l'authentification unilatérale de message (d'expéditeur) sont de la forme:

- *urn:liberty:security:2003-08:PEER:X509*, où PEER peut varier suivant le mécanisme d'authentification d'homologue déployé (par exemple néant, TLS, etc.).

Le mécanisme d'authentification de message WSS UIT-T X.509 repose sur [b-OASIS WSS X.509 profile] pour l'authentification de l'expéditeur du message auprès du destinataire. Ces mécanismes d'authentification de message sont unilatéraux. Autrement dit, seul l'expéditeur du message est authentifié. Il n'entre pas dans le cadre de la présente Recommandation de spécifier quand les messages de réponse doivent être authentifiés mais il convient de noter qu'on pourrait aussi utiliser ce mécanisme pour authentifier le message de réponse. Il est toutefois recommandé de reconnaître que dans le cas où l'authentification des messages de réponse est indépendante, la sémantique de protection de flux de message n'est pas la même que dans le cas d'un mécanisme d'authentification mutuelle d'entité homologue.

Dans le cas où les paramètres de déploiement nécessitent une authentification de message indépendante de l'authentification d'entité homologue, l'expéditeur est tenu, pour l'authentification de message, de prouver qu'il est en possession de la clé associée au jeton UIT-T X.509. Le destinataire doit reconnaître que cette clé appartient à l'expéditeur.

Lorsque l'expéditeur brandit la clé de confirmation du sujet pour signer des éléments du message, la signature garantit l'authenticité et l'intégrité des éléments qu'elle couvre. Toutefois, cette signature seule ne réduit pas les menaces de réexécutions, d'insertions et de certains types d'attaques visant à modifier le message. Pour protéger le message contre ces menaces, on peut recourir à l'un des mécanismes qui prennent en charge l'authentification de l'entité homologue ou au modèle de traitement de demande de lien SOAP sous-jacent.

i) Règles de traitement au niveau de l'expéditeur

Les règles énoncées dans le présent paragraphe viennent s'ajouter aux règles de traitement génériques pour l'authentification de message spécifiées dans la présente Recommandation.

- L'expéditeur est tenu de prouver qu'il est en possession de la clé privée associée à la signature générée conjointement avec le profil de jeton WSS UIT-T X.509.
- Dans le cas où les paramètres de déploiement nécessitent une authentification de message indépendante, pour accomplir l'obligation, il faut signer les parties appropriées du message et enregistrer les informations dans l'en-tête <wsse:Security> (comme décrit dans [b-OASIS WSS SOAP]).
- Dans le cas où les paramètres de déploiement ne nécessitent pas une authentification de message indépendante, l'expéditeur doit accomplir cette obligation en incluant dans l'en-tête de sécurité un élément <ds:KeyInfo> contenant le certificat.

Il faut pouvoir vérifier sans ambiguïté que cet élément correspond bien au certificat et à la clé utilisés pour établir l'authentification de l'entité homologue. Cela est nécessaire pour réduire la menace d'une attaque visant à remplacer un certificat. Il convient par ailleurs de noter que cette optimisation ne s'applique qu'aux mécanismes *ClientTLS:X509*.

ii) Règles de traitement au niveau du destinataire

- Si la politique de validation concernant l'authentification de l'entité homologue est suffisante aux fins d'authentification, le destinataire est tenu d'établir la correspondance entre d'une part le certificat et la clé utilisés pour établir l'authentification de l'homologue et d'autre part les informations de clé correspondantes acheminées dans le message. Cela permet au destinataire du message de déterminer que l'expéditeur du message prévoyait l'utilisation d'une identité authentifiée de transport particulière. Les informations reliant la clé SSL/TLS au message peuvent être acheminées dans le message en utilisant un jeton UIT-T X.509 de sécurité de message SOAP OASIS.

6.6 Protection des informations d'identification personnelle (PII)

D'après [UIT-T Y.2720], *Cadre de gestion d'identité dans les NGN*, la protection des informations PII doit faire l'objet de réglementations nationales et régionales. Si les mécanismes et procédures

employés pour prendre en charge la protection des informations PII peuvent varier suivant ces réglementations, ils reposent sur des principes communs.

Le présent paragraphe donne un bref aperçu des procédures de protection des informations PII qui sont spécifiées dans le rapport spécial du NIST *Guide to Protecting the Confidentiality of Personally Identifiable Information (PII)* [b-NIST-SP 800-122]. Les spécifications énoncées dans ce document relatives aux *sauvegardes de la confidentialité des informations PII* pourraient servir de guide aux concepteurs des systèmes IdM. Les catégories suivantes de sauvegardes sont définies:

- Sauvegardes opérationnelles:
  - création de politiques et de procédures;
  - sensibilisation, formation et éducation.
- Sauvegardes propres au respect de la vie privée:
  - réduire au strict minimum l'utilisation, la collecte et la conservation d'informations PII;
  - réaliser des évaluations d'incidence sur la vie privée;
  - désidentifier les informations;
  - rendre les informations anonymes.
- Contrôles de sécurité:

  Le paragraphe sur les contrôles de sécurité donne des indications sur les mécanismes et procédures de sécurité qui ne sont pas propres aux informations PII mais qui peuvent être employés pour la protection des informations PII. De même, les mécanismes de sécurité qui ne sont pas propres aux informations PII qui sont spécifiés dans [UIT-T Y.2704] peuvent être utilisés pour la protection des informations PII.

6.7 Fonctions d'identité fédérée

Dans [UIT-T Y.2721], *Spécifications et cas d'utilisation de la gestion d'identité dans les réseaux NGN*, on explique que *d'une manière générale, une fédération vise à permettre à chacun de ses membres de rester indépendant tout en facilitant l'échange d'informations d'identité spécifiques pour permettre des services fédérés.*

Dans le présent paragraphe, on recommande l'utilisation de deux mécanismes standards largement mis en œuvre qui permettent à un utilisateur d'accéder à de multiples services sans devoir s'abonner séparément à chaque service.

La Recommandation SAML [UIT-T X.1141] décrit une solution standard en matière de fédération, qui est principalement utilisée par les entreprises, les organismes publics et leurs fournisseurs de service.

La spécification OpenID [b-OpenID v.2] décrit une solution centrée sur l'utilisateur, qui est largement utilisée pour l'accès aux services web sur l'Internet.

6.7.1 Relais et interfonctionnement

La présente Recommandation décrit un certain nombre de mécanismes qui prennent en charge le relais et l'interfonctionnement entre différentes solutions IdM et fédérations. Les principaux mécanismes sont décrits dans les paragraphes suivants:

- Intégration de l'authentification basée sur l'infrastructure PKI avec le sous-système IMS (§ 6.2.6)
- Intégration de l'authentification basée sur l'infrastructure PKI avec des assertions SAML (§ 6.2.7)
- Intégration de l'authentification basée sur OpenID avec l'authentification AKA (§ 6.2.8)
- Architecture d'amorçage générique (§ 6.2.9)

- Corrélation et lien (§ 6.3)
- Fonctions d'identité fédérée (§ 6.7).

6.7.2 Découverte de fournisseurs IdSP dans un environnement fédéré

Le § 11.4.3 de la Recommandation SAML [UIT-T X.1141] définit le *profil de découverte de fournisseur d'identité*, qui permet à un fournisseur de service de découvrir les fournisseurs d'identité d'un utilisateur. Le profil est spécifié en appui au *profil SSO de navigateur web* SAML (défini au § 11.4.1 de [UIT-T X.1141]).

La spécification OpenID [b-OpenID v.2] spécifie un mécanisme de découverte qui permet à une partie utilisatrice de localiser un fournisseur IdSP d'utilisateur sur la base de l'identificateur OpenID fourni par l'utilisateur.

6.8 Contrôle d'accès aux informations d'identité

D'après [UIT-T Y.2721], les informations d'identité ne doivent être accessibles qu'aux entités autorisées sous réserve des règlementations et politiques applicables. Le présent paragraphe décrit des mécanismes qui peuvent être utilisés pour vérifier les privilèges d'autorisation.

6.8.1 Mécanisme de partage d'attributs basé sur SAML

Des assertions SAML contenant des déclarations d'attribut peuvent être utilisées comme mécanisme de gestion des privilèges. Le mécanisme décrit au § 6.2.1 peut être utilisé pour la distribution de jetons SAML.

6.8.2 Infrastructure de gestion des privilèges basée sur UIT-T X.509

Le cadre de certificat d'attribut défini dans [UIT-T X.509] peut être utilisé comme mécanisme pour une infrastructure de gestion des privilèges.

6.9 Authentification unique

L'authentification unique (SSO, *single sign-on*) est une capacité de réseau qui permet à un utilisateur de se connecter une seule fois et d'obtenir un accès aux multiples services d'application d'un réseau sans qu'il lui soit demandé de fournir des justificatifs d'authentification pour chaque service d'application. Cette capacité est très utile pour l'utilisateur en ce sens qu'elle lui permet de recevoir divers services sans avoir à conserver de multiples justificatifs d'authentification (par exemple des paires nom d'utilisateur/mot de passe). Etant donné que l'authentification unique permet à un utilisateur d'avoir un seul jeu de justificatifs d'authentification pour accéder à de multiples services d'application, il est plus facile pour les fournisseurs de service d'appliquer des règles strictes pour l'établissement des justificatifs, ce qui permet d'améliorer la sécurité du réseau.

Toutefois, si les justificatifs de l'utilisateur sont compromis, l'impact sur les réseaux utilisant l'authentification unique pourrait être plus grand que sur les systèmes qui ne prennent pas en charge l'authentification unique. Il est donc essentiel d'employer des mécanismes sécurisés pour l'authentification unique. Le présent paragraphe donne un aperçu de plusieurs mécanismes qui peuvent être utilisés pour prendre en charge l'authentification unique.

6.9.1 Mécanisme basé sur GBA

Le § 6.2.9 décrit l'utilisation de l'architecture GBA pour l'authentification d'un utilisateur auprès de n'importe quelle fonction d'application de réseau (NAF). De fait, l'architecture GBA offre un mécanisme unique permettant à un utilisateur de se connecter à toutes les fonctions NAF compatibles GBA sur un réseau. Si un utilisateur s'est connecté à une fonction NAF, la fonction BSF et l'équipement d'utilisateur se sont déjà mutuellement authentifiés et ont établi une clé partagée (Ks). La procédure de connexion de l'utilisateur à une autre fonction NAF comportera alors les étapes 1, 3, 4, 5 et 6 (l'étape 2 est sautée), qui sont décrites au § 6.2.9. A nouveau, la

procédure conduit au partage d'un secret (Ks_NAF) entre l'équipement d'utilisateur et la nouvelle fonction NAF. Ce secret partagé peut être utilisé pour l'authentification entre l'équipement d'utilisateur et la fonction NAF.

L'utilisation de l'authentification unique (SSO) basée sur GBA est recommandée dans les environnements dans lesquels l'architecture GBA a été déployée.

6.9.2 Mécanisme basé sur SAML

Les mécanismes d'authentification unique (SSO) basée sur SAML sont spécifiés au § 11.4, *Profils SSO de SAML*, de [UIT-T X.1141], qui définit un ensemble de profils SAML prenant en charge l'authentification unique, dont un *profil de déconnexion unique* (§ 11.4.4 de [UIT-T X.1141]). Ce profil spécifie une procédure qui permet à un utilisateur de se déconnecter de toutes les applications auxquelles il s'est connecté au moyen de l'authentification unique.

La spécification SAML v.2 repose sur le fait que des relations de confiance ont été établies à l'avance entre un fournisseur IdSP et des parties utilisatrices. Elle prend par ailleurs en charge le fait que des identificateurs pseudonymes sont disponibles entre un fournisseur IdSP et une partie utilisatrice. Elle est adaptée aux applications pour lesquelles un accord contractuel (par exemple SLA) a été conclu ou pour lesquelles les informations ou les transactions en jeu ont une grande valeur.

6.9.3 Mécanisme basé sur OpenID

L'authentification OpenID 2.0 prend en charge la capacité d'authentification unique afin de permettre à un utilisateur final d'accéder à plusieurs parties utilisatrices une fois qu'il a été authentifié avec succès. Elle ne nécessite pas l'existence d'une relation de confiance entre fournisseur IdSP et partie utilisatrice. Etant donné qu'elle prend uniquement en charge le format URL-URI pour identifier les utilisateurs, un système DNS est nécessaire pour son utilisation. Elle est donc adaptée aux applications de services web pour lesquelles les informations et les transactions en jeu ont relativement peu de valeur.

6.10 Déconnexion unique

Le *protocole de déconnexion unique* SAML, décrit au § 8.2.7 de [UIT-T X.1141], permet à l'utilisateur final de se déconnecter presque simultanément des multiples sessions auxquelles il participe. Ces sessions sont celles qui ont été établies par le biais du fournisseur IdSP (sessions entre l'utilisateur et des applications pour lesquelles le fournisseur IdSP a asserté l'identité de l'utilisateur). Le fournisseur IdSP suit toutes les sessions authentifiées avec diverses parties utilisatrices que l'utilisateur a établies par le biais du fournisseur IdSP. En particulier, les justificatifs d'authentification (par exemples cookies, assertions) doivent être invalidés pour les sessions auxquelles il est mis fin. Le protocole peut être utilisé dans les cas suivants:

1) l'utilisateur se déconnecte de l'une des sessions et indique qu'il souhaite se déconnecter de toutes les sessions qui ont été lancées par le fournisseur IdSP;
2) l'utilisateur indique directement au fournisseur IdSP qu'il souhaite se déconnecter de toutes les sessions;
3) le fournisseur IdSP déconnecte un utilisateur sans que celui-ci n'en ait fait la demande (par exemple par suite de l'expiration d'une temporisation).

Le protocole définit les entités participantes, leur comportement, le flux de messages et le format des messages échangés. Les sous-paragraphes qui suivent décrivent l'utilisation du protocole dans les cas énumérés ci-dessus.

6.10.1 L'utilisateur se déconnecte de l'une des sessions et indique qu'il souhaite se déconnecter de toutes les sessions qui ont été lancées par le fournisseur IdSP

La Figure illustre les étapes de base du flux de messages, qui sont décrites ci-dessous.

6.10.1.1 Entités intervenant dans la procédure et le flux d'information

Ces entités sont les suivantes:

- fonction d'utilisateur final;
- serveur d'application 1 (AS1): entité fournissant un service, qui joue le rôle de partie utilisatrice et fait office de demandeur et de répondeur SAML, comme défini dans [UIT-T X.1141];
- serveur d'application 2 (AS2): entité fournissant un service, qui joue le rôle de partie utilisatrice et fait office de demandeur et de répondeur SAML, comme défini dans [UIT-T X.1141];
- entité A-2: entité fonctionnelle de passerelle d'application (APL-GW-FE), qui joue le rôle de fournisseur IdSP et fait office de demandeur et de répondeur SAML, comme défini dans [UIT-T X.1141].

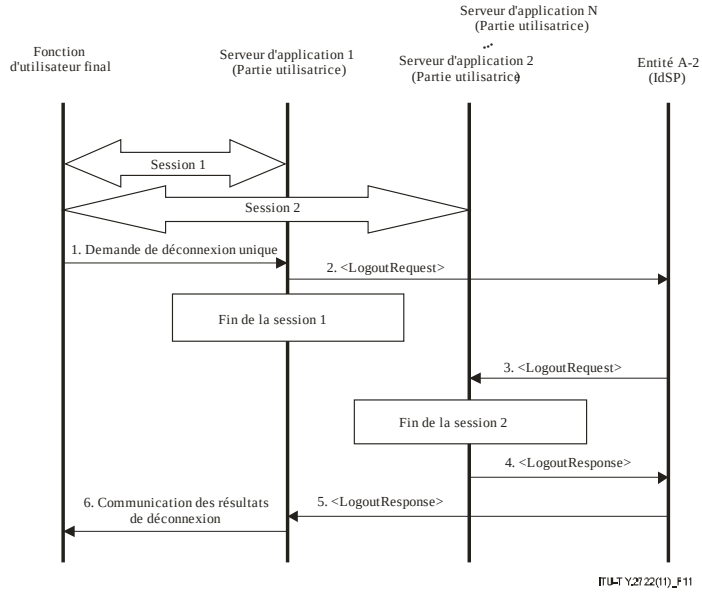


Figure 11 – Déconnexion unique basée sur SAML demandée par un utilisateur à une session à laquelle il participe

Les étapes de base de la procédure de déconnexion unique sont les suivantes:

1) La fonction d'utilisateur final envoie une demande de déconnexion au serveur d'application 1 (AS1) indiquant qu'il souhaite se déconnecter de toutes les sessions auxquelles il participe.

2) Le serveur AS1 demande une déconnexion de toutes les sessions auxquelles l'utilisateur participe en envoyant une demande <LogoutRequest> à l'entité A-2. La demande devrait être signée dans un souci d'authentification et de protection de l'intégrité, comme spécifié au § 8.2.7 de [UIT-T X.1141].

   Après cette étape, le serveur AS1 tente de mettre fin à la session 1. Pour cela, il invalide les justificatifs d'authentification de la session (par exemple assertions, cookies), ce qui obligera la fonction d'utilisateur final à passer par une procédure d'authentification, si elle envoie une autre demande au serveur AS1.

3) Après avoir validé la demande provenant du serveur AS1, l'entité A-2 envoie des messages <LogoutRequest> à toutes les parties utilisatrices (seul le serveur AS2 est illustré sur la Figure 11). Les demandes devraient être signées, comme spécifié au § 8.2.7 de [UIT-T X.1141].

Après avoir validé la demande de déconnexion, le serveur AS2 tente de mettre fin à la session 2.

4) Le serveur AS2 communique à l'expéditeur de la demande de déconnexion (entité A-2) le résultat de la tentative de déconnexion en envoyant une réponse <LogoutResponse>, qui devrait être signée.

5) L'entité A-2 envoie une réponse <LogoutResponse> à l'expéditeur initial de la demande de déconnexion (serveur AS1) pour lui communiquer les résultats de la déconnexion unique (par exemple réussite, déconnexion partielle). La réponse devrait être signée.

   Après cette étape, l'entité A-2 met à jour sa liste de sessions actives et invalide les justificatifs d'authentification (par exemple cookies, assertions) pour les sessions auxquelles il devait être mis fin.

6) Le serveur AS1 communique le résultat de déconnexion en réponse à la demande envoyée par la fonction d'utilisateur final à l'étape 1.

6.10.2 L'utilisateur indique directement au fournisseur IdSP qu'il souhaite se déconnecter de toutes les sessions

La Figure 12 illustre les étapes de base du flux de messages, qui sont décrites ci-dessous.

6.10.2.1 Entités intervenant dans la procédure et le flux d'information

Les entités intervenant dans la procédure de déconnexion sont les mêmes que celles qui sont décrites au § 6.10.1.1. Dans le présent cas d'utilisation, la fonction d'utilisateur final a une session à part (session 3) avec l'entité A-2 (IdSP). Elle utilise cette session pour envoyer une demande de déconnexion unique à l'étape 1. Les autres étapes de la procédure sont analogues à celles qui sont décrites au § 6.10.1.1.

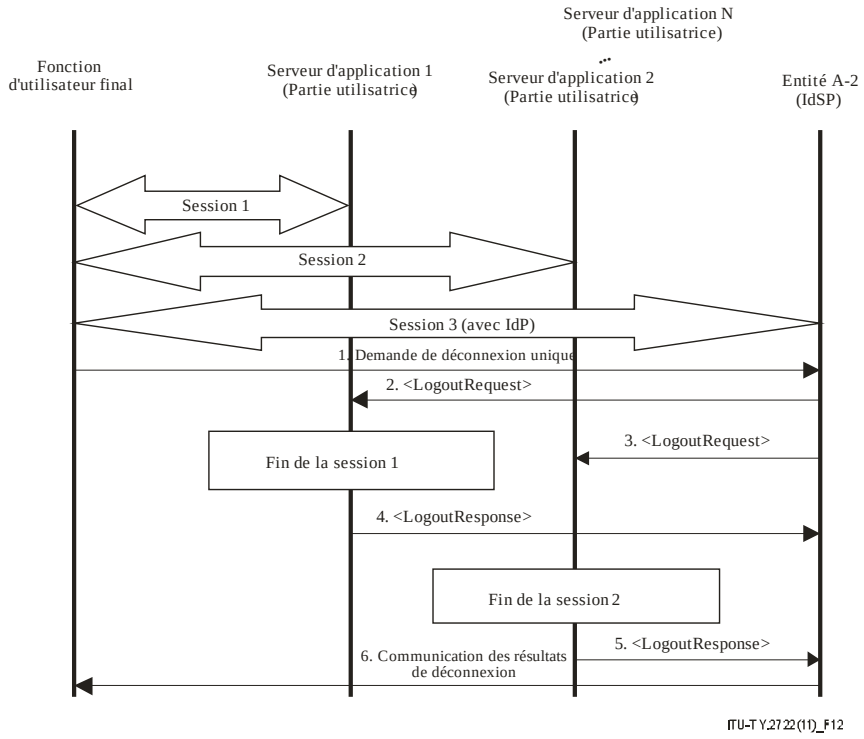


Figure 12 – Déconnexion unique basée sur SAML demandée par un utilisateur au fournisseur IdSP

Les étapes de base de la procédure de déconnexion unique sont les suivantes:

1) La fonction d'utilisateur final envoie une demande de déconnexion unique directement à l'entité A-2.

2) L'entité A-2 envoie une demande <LogoutRequest> au serveur AS1. La demande devrait être signée dans un souci d'authentification et de protection de l'intégrité, comme spécifié au § 8.2.7 de [UIT-T X.1141].

   Après avoir validé la demande, le serveur AS1 tente de mettre fin à la session 1. Pour cela, il invalide les justificatifs d'authentification de la session (par exemple assertions, cookies), ce qui obligera la fonction d'utilisateur final à passer par une procédure d'authentification, s'il envoie une autre demande au serveur AS1.

3) L'entité A-2 envoie une demande <LogoutRequest> au serveur AS2 (elle envoie également la demande à tous les autres serveurs des sessions auxquelles l'utilisateur participe). Cette étape est analogue à l'étape 2.

   Après avoir validé la demande de déconnexion, le serveur AS2 tente de mettre fin à la session 2.

4) Le serveur AS1 communique à l'expéditeur de la demande de déconnexion (entité A-2) le résultat de la tentative de déconnexion en envoyant une réponse <LogoutResponse>, qui devrait être signée.

5) De manière analogue à l'étape précédente, le serveur AS2 communique à l'expéditeur de la demande de déconnexion (entité A-2) le résultat de la tentative de déconnexion en envoyant une réponse <LogoutResponse> signée.

   Après cette étape, l'entité A-2 met à jour sa liste de sessions actives et invalide les justificatifs d'authentification (par exemple cookies, assertions) pour les sessions auxquelles il devait être mis fin.

Après avoir validé toutes les réponses de déconnexion, l'entité A-2 communique à la fonction d'utilisateur final le résultat de la déconnexion unique. Il s'agit de la réponse à la demande envoyée par la fonction d'utilisateur final à l'étape 1.

7 Sécurité

Les mécanismes traités dans la présente Recommandation et les mécanismes spécifiés dans [UIT-T Y.2704] répondent aux exigences de sécurité IdM spécifiées dans [UIT-T Y.2721].

Appendice I

Authentification de message WSS UIT-T X.509 v3

(Cet appendice ne fait pas partie intégrante de la présente Recommandation.)

On trouvera ci-après un exemple de traitement des messages avec un jeton WSS UIT-T X.509, comme décrit au § 6.5.1.2.

```
<?xml version="1.0" encoding="UTF-8"?>
<s:Envelope xmlns:s="http://schemas.xmlsoap.org /soap/envelope/"
      xmlns:sb="urn:liberty:sb:2006-08"
      xmlns:pp="urn:liberty:id-sis-pp:2003-08"
      xmlns:sec="urn:liberty:security:20 06-08"
      xmlns:wsse="http://docs.oasis-open.org/wss/2004 /01/oasis-200401-wss-wssecurity-
      secext-1.0.xsd"
      xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-
      utility-1.0.xsd"
      xmlns:wsa="http://www.w3.org/2005/08/addressing">

<s:Header>
<!-- see Liberty SOAP Binding Specification for which headers are required and optional -
->

<wsa:MessageID wsu:Id="mid">...</wsa:MessageID>

<wsa:To wsu:Id="to">...</wsa:To>

<wsa:Action wsu:Id="action">...</wsa:Action>

<wsse:Security mustUnderstand="1">

       <wsu:Timestamp wsu:Id="ts">
       <wsu:Created>2005-06-17T04:49:17Z</ wsu:Created >
       </wsu:Timestamp>

       <wsse:BinarySecurityToken
              ValueType="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-x509-
token-profile-1.0#X509v3 "
              wsu:Id="X509Token"
              EncodingType="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-soap-
message-security-1.0#Base64Binar y">
              MIIB9zCCAWSgAwIBAgIQ...
       </wsse:BinarySecurityToken>

       <ds:Signature xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
              <ds:SignedInfo>

                     <!-- in general include a ds:Reference for each wsa: header added
according to SOAP binding -->

                     <!-- include the MessageID in the signature -->
                     <ds:Reference URI="#mid">...</ds:Reference>

                     <!-- include the To in the signature -->
                     <ds:Reference URI="#to">...</ds:Reference>

                     <!-- include the Action in the signature -->
                     <ds:Reference URI="#action">...</ds:Reference>

                     <!-- include the Timestamp in the signature -->
                     <ds:Reference URI="#ts">...</ds:Reference>

                     <!-- bind the security token (thwart cert substitution attacks) -->
                     <ds:Reference URI="#X509Token">
                            <ds:DigestMethod Algorithm="http://www.w3.org/2000/09/x
mldsig#sha1"/>
                            <ds:DigestValue>Ru4cAfeBABE...</ ds:DigestValue>
```

```
                    </ds:Reference>

                    <!-- bind the body of the message -->
                    <ds:Reference URI="#MsgBody">
                           <ds:DigestMethod Algorithm="http://www.w3.org/2000/09/
                           xmldsig# sha1"/>
                           <ds:DigestValue>YgGfS0pi56pu...</ds:DigestValue>
                    </ds:Reference>
             </ds:SignedInfo>
             <ds:KeyInfo>
                    <wsse:SecurityTokenReference>
                           <wsse:Reference URI="#X509Token" />
                    </wsse:SecurityTokenReference>
             </ds:KeyInfo>
             <ds:SignatureValue>
                    HJJWbvqW9E84vJVQkjjLLA6nNvBX7mY00TZhwBdFNDElgscS XZ5Ekw==
             </ds:SignatureValue>
       </ds:Signature>
</wsse:Security>
</s:Header>

<s:Body wsu:Id="MsgBody">
       <pp:Modify>
             <!-- this is an ID-SIS-PP Modify message -->
       </pp:Modify>
</s:Body>


</s:Envelope>
```

Appendice II

Mécanisme "OpenID + OAuth" pour le contrôle d'accès

(Cet appendice ne fait pas partie intégrante de la présente Recommandation.)

Le § 6.2.8 décrit l'utilisation de OpenID pour l'authentification d'un utilisateur auprès d'une fonction d'application de réseau (NAF). Il est proposé ici d'introduire OAuth basé sur OpenID pour la protection des informations PII et le contrôle d'accès.

II.1 OAuth ([b-IETF RFC 5849])

OAuth est un protocole ouvert permettant à une application d'accéder aux informations relatives à un utilisateur final auprès d'un service web lorsque l'application est autorisée par l'utilisateur final. Les informations relatives à l'utilisateur final sont transférées en toute sécurité sans révéler l'identité de l'utilisateur.

OAuth a pour objectif d'obtenir auprès du serveur web un jeton d'accès, qui peut ensuite être utilisé pour échanger des données propres à l'utilisateur avec un service web (par exemple des informations de calendrier ou un répertoire d'adresses). La procédure OAuth normale est une séquence à quatre étapes:

1) Demander un jeton de "demande".
2) Demander que le jeton soit autorisé, ce qui entraîne l'approbation par l'utilisateur.
3) Echanger le jeton de demande autorisé contre un jeton d'"accès".
4) Utiliser le jeton d'accès pour interagir avec les données de service web relatives à l'utilisateur.

Pour en savoir plus sur le protocole OAuth, on se reportera à [b-IETF RFC 5849].

II.2 Utilisation de OpenID conjointement avec OAuth

Tandis que OpenID peut être utilisé comme mécanisme de gestion d'identité pour authentifier des utilisateurs, OAuth pourrait également être utilisé pour autoriser l'accès à des données d'utilisateur sensibles. Dans un tel scénario, le fournisseur IdM assure des fonctions combinées et fait office à la fois de fournisseur d'identité OpenID et de fournisseur de service OAuth.

II.3 Flux d'autorisation OpenID + OAuth

Avec OpenID + OAuth, la séquence reste essentiellement la même. La différence est que l'obtention d'un jeton de demande OAuth autorisé (étapes 1 et 2) est incluse dans la demande d'authentification OpenID. De cette façon, l'utilisateur peut approuver simultanément une connexion et un accès aux services.

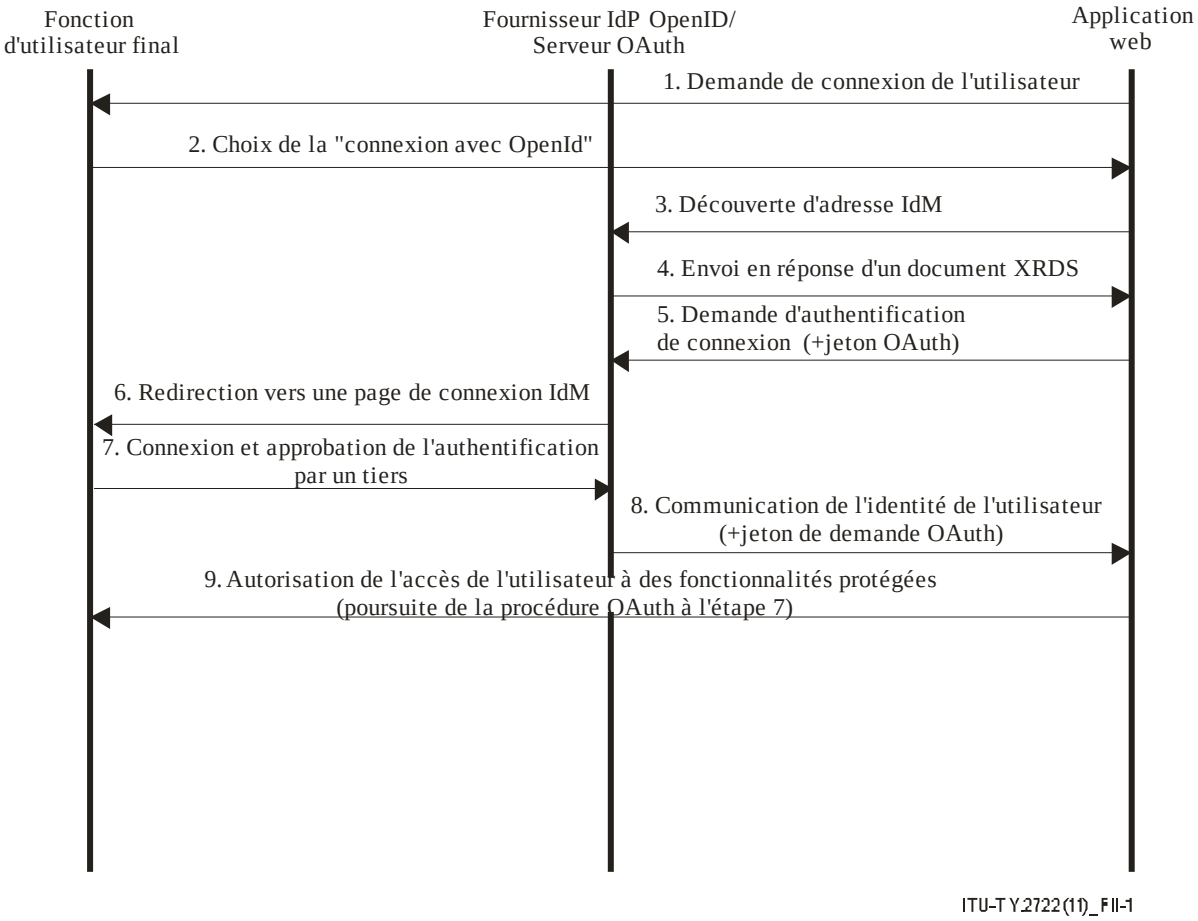


Figure II.1 – Authentification basée sur OpenID + OAuth

Les étapes de base sont les suivantes:

1) L'application web demande à l'utilisateur final de se connecter en lui offrant un ensemble d'options de connexion, y compris l'utilisation de son compte OpenID.

2) L'utilisateur choisit l'option de "connexion avec OpenID".

3) L'application web envoie une demande de "découverte" au fournisseur IdSP pour obtenir des informations relatives au point d'extrémité d'authentification de connexion IdSP.

4) Le fournisseur IdSP retourne un document XRDS, qui contient l'adresse du point d'extrémité.

5) L'application web envoie une demande d'authentification de connexion à l'adresse du point d'extrémité IdSP.

6) Cette action redirige l'utilisateur vers une page de connexion fédérée IdSP, soit dans la même fenêtre du navigateur soit dans une fenêtre incrustée, et il est demandé à l'utilisateur de se connecter.

7) Une fois l'utilisateur connecté, le fournisseur IdSP affiche une page de confirmation et indique à l'utilisateur qu'une application tierce demande une authentification. Sur la page, il est demandé à l'utilisateur de confirmer ou de rejeter l'établissement d'un lien entre la connexion au compte IdSP et la connexion à l'application web. Il est ensuite demandé à l'utilisateur d'approuver l'accès à un ensemble spécifié de services IdSP. Pour que l'authentification puisse se poursuivre, l'utilisateur doit approuver à la fois la connexion et le partage des informations le concernant.

8) Si l'utilisateur approuve l'authentification, le fournisseur IdSP retourne des informations relatives à l'utilisateur à l'URL spécifiée dans le paramètre `openid.return_to` de la demande initiale. Un identificateur fourni par le fournisseur IdSP, qui n'a aucun lien avec le nom ou

le mot de passe associé au compte IdM actuel de l'utilisateur, est joint en tant que paramètre de requête `openid.claimed_id`. Si la demande portait aussi sur un échange d'attributs, d'autres informations relatives à l'utilisateur peuvent être jointes. Concernant OpenID + OAuth, un jeton de demande OAuth autorisé est également retourné.

9) L'application web utilise l'identificateur fourni par le fournisseur IdSP pour reconnaître l'utilisateur et autoriser l'accès aux fonctionnalités et aux données de l'application. Concernant OpenID+OAuth, l'application web utilise le jeton de demande pour poursuivre la séquence OAuth et obtenir un accès aux services IdSP de l'utilisateur.

Bibliographie

[b-ETSI TS 133 220] ETSI TS 133 220 V6.3.0 (2004), *Universal Mobile Telecommunications System (UMTS); Generic Authentication Architecture (GAA); Generic bootstrapping architecture*.

[b-IETF RFC 2616] IETF RFC 2616 (1999), *Hypertext Transfer Protocol – HTTP/1.1*. <http://datatracker.ietf.org/doc/rfc2616/>

[b-IETF RFC 2617] IETF RFC 2617 (1999), *HTTP Authentication: Basic and Digest Access Authentication*. <http://datatracker.ietf.org/doc/rfc2617/>

[b-IETF RFC 3310] IETF RFC 3310 (2002), *Hypertext Transfer Protocol (HTTP) Digest Authentication Using Authentication and Key Agreement (AKA)*. <http://www.rfc-editor.org/rfc/rfc3310.txt>

[b-IETF RFC 4169] IETF RFC 4169 (2005), *Hypertext Transfer Protocol (HTTP) Digest Authentication Using Authentication and Key Agreement (AKA) Version-2*. <http://www.ietf.org/rfc/rfc4169.txt?number=4169>

[b-IETF RFC 4279] IETF RFC 4279 (2005), *Pre-Shared Key Ciphersuites for Transport Layer Security (TLS)*. <http://datatracker.ietf.org/doc/rfc4279/>

[b-IETF RFC 5849] IETF RFC 5849 (2010), *The OAuth 1.0 Protocol*. <http://tools.ietf.org/html/rfc5849>

[b-LA WSF] Liberty Alliance (2008), *Web Services Framework: A Technical Overview*. <http://www.projectliberty.org/liberty/content/download/4120/27687/file/idwsf-intro-v1.0.pdf>

[b-LA ID-WSF security] Liberty Alliance Project (2007), *Liberty ID-WSF Security Mechanisms Core version 2.0-errata version 1.0*.

[b-LA SOAP binding] Liberty Alliance Project Web Services Security (WSS) (2006), *Liberty SOAP Binding Version 2.0*.

[b-NIST-SP 800-122] NIST Special Publication SP 800-122 (2010), *Guide to Protecting the Confidentiality of Personally Identifiable Information (PII)*. <http://csrc.nist.gov/publications/nistpubs/800-122/sp800-122.pdf>

[b-OASIS SAML token] OASIS (2006), *Web Services security: SAML Token Profile 1.1, and its Approved Errata 1*.

[b-OASIS WSS SOAP] OASIS (2004), *Web Services Security: SOAP Message Security 1.1 (WS-Security 2004)*.

[b-OASIS WSS X.509 profile] OASIS (2006), *Web Services Security X.509 Certificate Token Profile 1.1*.

[b-OpenID v.2] *OpenID Authentication 2.0*. <http://openid.net/specs/openid-authentication-2_0.html>

[b-W3C XML signature] World Wide Web Consortium (W3C) (2008), *XML Signature Syntax and Processing (second edition)*.

[b-3GPP TR 33.924] 3GPP TR 33.924 Release 9 (2009), 3rd Generation Partnership Project, *Identity management and 3GPP security interworking; Identity management and Generic Authentication Architecture (GAA) interworking (Release 9)*.

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	Terminaux et méthodes d'évaluation subjectives et objectives
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, 2011