МЕЖДУНАРОДНЫЙ СОЮЗ ЭЛЕКТРОСВЯЗИ

# МСЭ-Т X.1141

СЕКТОР СТАНДАРТИЗАЦИИ
ЭЛЕКТРОСВЯЗИ МСЭ

(06/2006)

СЕРИЯ X: СЕТИ ПЕРЕДАЧИ ДАННЫХ, ВЗАИМОСВЯЗЬ ОТКРЫТЫХ СИСТЕМ И БЕЗОПАСНОСТЬ

Безопасность электросвязи

# Язык разметки, предусматривающий защиту данных (SAML 2.0)

Рекомендация МСЭ-Т X.1141

РЕКОМЕНДАЦИИ МСЭ-Т СЕРИИ X

**СЕТИ ПЕРЕДАЧИ ДАННЫХ, ВЗАИМОСВЯЗЬ ОТКРЫТЫХ СИСТЕМ И БЕЗОПАСНОСТЬ**

| | |
|---|---|
| СЕТИ ПЕРЕДАЧИ ДАННЫХ ОБЩЕГО ПОЛЬЗОВАНИЯ | |
| Службы и услуги | X.1–X.19 |
| Интерфейсы | X.20–X.49 |
| Передача, сигнализация и коммутация | X.50–X.89 |
| Сетевые аспекты | X.90–X.149 |
| Техническое обслуживание | X.150–X.179 |
| Административные предписания | X.180–X.199 |
| ВЗАИМОСВЯЗЬ ОТКРЫТЫХ СИСТЕМ | |
| Модель и обозначение | X.200–X.209 |
| Определения служб | X.210–X.219 |
| Спецификации протоколов с установлением соединений | X.220–X.229 |
| Спецификации протоколов без установления соединений | X.230–X.239 |
| Проформы PICS | X.240–X.259 |
| Идентификация протоколов | X.260–X.269 |
| Протоколы обеспечения безопасности | X.270–X.279 |
| Управляемые объекты уровня | X.280–X.289 |
| Испытание на соответствие | X.290–X.299 |
| ВЗАИМОДЕЙСТВИЕ МЕЖДУ СЕТЯМИ | |
| Общие положения | X.300–X.349 |
| Спутниковые системы передачи данных | X.350–X.369 |
| Сети, основанные на протоколе Интернет | X.370–X.379 |
| СИСТЕМЫ ОБРАБОТКИ СООБЩЕНИЙ | X.400–X.499 |
| СПРАВОЧНИК | X.500–X.599 |
| ОРГАНИЗАЦИЯ СЕТИ ВОС И СИСТЕМНЫЕ АСПЕКТЫ | |
| Организация сети | X.600–X.629 |
| Эффективность | X.630–X.639 |
| Качество обслуживания | X.640–X.649 |
| Наименование, адресация и регистрация | X.650–X.679 |
| Абстрактно-синтаксическая нотация 1 (ASN.1) | X.680–X.699 |
| УПРАВЛЕНИЕ В ВОС | |
| Структура и архитектура управления системами | X.700–X.709 |
| Служба и протокол связи для общего управления | X.710–X.719 |
| Структура управляющей информации | X.720–X.729 |
| Функции общего управления и функции ODMA | X.730–X.799 |
| БЕЗОПАСНОСТЬ | X.800–X.849 |
| ПРИЛОЖЕНИЯ ВОС | |
| Фиксация, параллельность и восстановление | X.850–X.859 |
| Обработка транзакций | X.860–X.879 |
| Удаленные операции | X.880–X.889 |
| Общие приложения ASN.1 | X.890–X.899 |
| ОТКРЫТАЯ РАСПРЕДЕЛЕННАЯ ОБРАБОТКА | X.900–X.999 |
| **БЕЗОПАСНОСТЬ ЭЛЕКТРОСВЯЗИ** | **X.1000–** |

*Для получения более подробной информации просьба обращаться к перечню Рекомендаций МСЭ-Т.*

# Рекомендация МСЭ-Т X.1141

## Язык разметки, предусматривающий защиту данных (SAML 2.0)


### Резюме

SAML – это интегрированная среда XML для передачи информации о безопасности. Эта информация о безопасности выражается в форме подтверждений относительно субъектов, где субъект представляет собой элемент (человека или компьютер), идентифицированный в каком-либо домене безопасности. Отдельное подтверждение может содержать несколько различных внутренних утверждений относительно аутентификации, авторизации и атрибутов. В настоящей Рекомендации определяется протокол, при помощи которого клиент может запросить подтверждения от ответственных органов SAML и получить ответ от них. Этот протокол, состоящий из форматов сообщений запросов и ответов XML, может быть связан с множеством различных базовых протоколов связи и транспортировки; в настоящее время SAML определяет одну связь SOAP поверх HTTP. Создавая свои ответы, ответственные органы SAML могут использовать различные источники информации, например, внешние хранилища правил, и подтверждения, которые были получены в качестве исходных данных для запросов. В настоящей Рекомендации определяются элементы подтверждений SAML, объекты, условия, правила обработки и утверждения. Кроме того, в ней описан расширенный профиль Метаданных SAML, который содержит соответствующую область имен, общие типы данных, правила обработки и обработку подписи. Описано также несколько различных связей протокола, в частности, помимо всех других описаны связи SOAP, PAOS (обратная SOAP), HTTP Redirect, HTTP POST. В настоящей Рекомендации приводится исчерпывающий список профилей SAML, таких как профиль SSO веб-браузера и профиль единого выхода из системы, которые позволяют широко применять SAML 2.0 в промышленности. Приведены также рекомендации по вопросам аутентификации.

Настоящая Рекомендация технически эквивалентна и полностью совместима со стандартом OASIS SAML 2.0.



### Источник

Рекомендация МСЭ-Т X.1141 подтверждена 13 июня 2006 года 17-й Исследовательской комиссией МСЭ-Т (2005–2008 гг.) в соответствии с процедурой, изложенной в Рекомендации МСЭ-Т A.8.

## ПРЕДИСЛОВИЕ

Международный союз электросвязи (МСЭ) является специализированным учреждением Организации Объединенных Наций в области электросвязи. Сектор стандартизации электросвязи МСЭ (МСЭ-Т) – постоянный орган МСЭ. МСЭ-Т отвечает за изучение технических, эксплуатационных и тарифных вопросов и за выпуск Рекомендаций по ним с целью стандартизации электросвязи на всемирной основе.

На Всемирной ассамблее по стандартизации электросвязи (ВАСЭ), которая проводится каждые четыре года, определяются темы для изучения Исследовательскими комиссиями МСЭ-Т, которые, в свою очередь, вырабатывают Рекомендации по этим темам.

Утверждение Рекомендаций МСЭ-Т осуществляется в соответствии с процедурой, изложенной в Резолюции 1 ВАСЭ.

В некоторых областях информационных технологий, которые входят в компетенцию МСЭ-Т, необходимые стандарты разрабатываются на основе сотрудничества с ИСО и МЭК.

## ПРИМЕЧАНИЕ

В настоящей Рекомендации термин "администрация" используется для краткости и обозначает как администрацию электросвязи, так и признанную эксплуатационную организацию.

Соблюдение положений данной Рекомендации носит добровольный характер. Однако в Рекомендации могут содержаться определенные обязательные положения (например, для обеспечения возможности взаимодействия или применимости), и соблюдение положений данной Рекомендации достигается в случае выполнения всех этих обязательных положений. Для выражения необходимости выполнения требований используется синтаксис долженствования и соответствующие слова (такие, как "должен" и т. п.), а также их отрицательные эквиваленты. Использование этих слов не предполагает, что соблюдение положений данной Рекомендации является обязательным для какой-либо из сторон.

## ПРАВА ИНТЕЛЛЕКТУАЛЬНОЙ СОБСТВЕННОСТИ

МСЭ обращает внимание на вероятность того, что практическое применение или реализация этой Рекомендации может включать использование заявленного права интеллектуальной собственности. МСЭ не занимает какую бы то ни было позицию относительно подтверждения, обоснованности или применимости заявленных прав интеллектуальной собственности, независимо от того, отстаиваются ли они членами МСЭ или другими сторонами вне процесса подготовки Рекомендации.

На момент утверждения настоящей Рекомендации МСЭ не получил извещение об интеллектуальной собственности, защищенной патентами, которые могут потребоваться для выполнения этой Рекомендации. Однако те, кто будет применять Рекомендацию, должны иметь в виду, что это может не отражать самую последнюю информацию, и поэтому им настоятельно рекомендуется обращаться к патентной базе данных БСЭ по адресу: http://www.itu.int/ITU-T/ipr/.

**СОДЕРЖАНИЕ**

# Рекомендация МСЭ-Т X.1141

## Язык разметки, предусматривающий защиту данных (SAML 2.0)

### 1 Сфера применения

В настоящей Рекомендации определяется Язык разметки, предусматривающий защиту данных (SAML 2.0). Язык SAML определяет синтаксис и семантику обработки подтверждений, сделанных элементом системы относительно субъекта. В процессе формулирования или применения таких подтверждений, Элементы системы SAML могут использовать другие протоколы для связи либо с самим подтверждением, либо с субъектом подтверждения. В настоящей Рекомендации, в дополнение к правилам обработки, используемых при управлении системой SAML, определяется структура Подтверждений языка SAML, связанный с ними набор протоколов.

Подтверждения SAML и протокольные сообщения кодируются в виде XML и используют области имен XML. Как правило, для транспортировки они встраиваются в другие структуры, например, запросы HTTP POST или SOAP сообщения в кодировке XML. В настоящей Рекомендации определяются также связи языка SAML, которые обеспечивают возможность для встраивания и транспортировки протокольных сообщений языка SAML. Кроме того, в настоящей Рекомендации определяется базовый набор профилей для применения Подтверждений и протоколов SAML в особых условиях использования или для обеспечения взаимодействия в случаях применения возможностей SAML.

В настоящей Рекомендации определяется следующее:

1) требования по соответствию для SAML;
2) подтверждения и протоколы для языка SAML:
   - Схема SAML – Подтверждения;
   - Схема SAML – Протоколы;
3) связи для SAML;
4) профили для SAML:
   - Схема SAML – Профиль ECP;
   - Схема SAML – Профиль атрибута X.500/LDAP;
   - Схема SAML – Профиль атрибута DCE PAC;
   - Схема SAML – Профиль атрибута XACML.
5) метаданные для языка SAML;
6) Схема SAML – Метаданные;
7) контекст аутентификации для языка SAML.

### 2 Справочные документы

В нижеследующих Рекомендациях МСЭ-Т и других справочных документах содержатся положения, которые посредством ссылок в настоящем тексте составляют положения настоящей Рекомендации. На время публикации указанные здесь издания были действительными. Все рекомендации и другие справочные документы постоянно пересматриваются; поэтому всем пользователям данной Рекомендации настоятельно рекомендуется изучить возможность использования последних изданий перечисленных ниже рекомендаций и других справочных документов. Сектор стандартизации МСЭ поддерживает перечень рекомендаций МСЭ-Т, действующих на текущее время. Рабочая группа по стандартам для сети Internet (IETF) поддерживает перечень текущих Запросов для комментариев (RFC), а также тех RFC, которые были отменены более поздними их версиями. WWW-консорциум (W3C), Консорциум уникальных имен (Unicode Consortium) и Liberty Alliance поддерживают перечень последних рекомендаций и других публикаций.

- ITU-T Recommendation X.660 (2004) | ISO/IEC 9834-1:2005, *Information technology – Open Systems Interconnection – Procedures for the operation of OSI Registration Authorities: General procedures and top arcs of the ASN.1 Object Identifier tree*.
- ITU-T Recommendation X.667 (2004) | ISO/IEC 9834-8:2005, *Information technology – Open Systems Interconnection – Procedures for the operation of OSI Registration Authorities: Generation and Registration of Universally Unique Identifiers (UUIDs) and their use as ASN.1 Object Identifier components.*

- ITU-T Recommendation X.680 (2002) | ISO/IEC 8824-1:2002, *Information technology – Abstract Syntax Notation One (ASN.1): Specification of basic notation.*
- ITU-T Recommendation X.800 (1991), *Security architecture for Open Systems Interconnection for CCITT applications.*
- ITU-T Recommendation X.811 (1995) | ISO/IEC 10181-2:1996, *Information technology – Open Systems Interconnection – Security frameworks for open systems: authentication framework.*
- ITU-T Recommendation X.812 (1995) | ISO/IEC 10181-3:1996, *Information technology – Open Systems Interconnection – Security frameworks for open systems: Access control framework.*
- ITU-T Recommendation X.1142 (2006), *eXtensible Access Control Markup Language (XACML 2.0).*
- IETF RFC 1034 (1987), *Domain Names – Concepts and Facilities*.
- IETF RFC 1510 (1993), *The Kerberos Network Authentication Service (V5)*.
- IETF RFC 1750 (1994), *Randomness Recommendations for Security*.
- IETF RFC 1951 (1996), *DEFLATE Compressed Data Format Specification Version 1.3*.
- IETF RFC 1991 (1996), *PGP Message Exchange Formats*.
- IETF RFC 2045 (1996), *Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies*.
- IETF RFC 2119 (1997), *Keywords for use in RFCs to Indicate Requirement Levels*.
- IETF RFC 2246 (1999), *The TLS Protocol Version 1.0*.
- IETF RFC 2253 (1997), *Lightweight Directory Access Protocol (v3): UTF-8 String Representation of Distinguished Names*.
- IETF RFC 2396 (1998), *Uniform Resource Identifiers (URI): Generic Syntax*.
- IETF RFC 2535 (1999), *Domain Name System Security Extensions*.
- IETF RFC 2616 (1999), *Hypertext Transfer Protocol – HTTP/1.1*.
- IETF RFC 2617 (1999), *HTTP Authentication: Basic and Digest Access Authentication*.
- IETF RFC 2798 (2000), *Definition of the inetOrgPerson LDAP Object Class*.
- IETF RFC 2828 (2000), *Internet Security Glossary*.
- IETF RFC 2914 (2000), *Congestion Control Principles*.
- IETF RFC 2915 (2000), *The Naming Authority Pointer (NAPTR) DNS Resource Record*.
- IETF RFC 2945 (2000), *The SRP Authentication and Key Exchange System*.
- IETF RFC 2965 (2000), *HTTP State Management Mechanism*.
- IETF RFC 3023 (2001), *XML Media Types*.
- IETF RFC 3061 (2001), *A URN Namespace of Object Identifiers*.
- IETF RFC 3075 (2001), *XML-Signature Syntax and Processing*.
- IETF RFC 3377 (2002), *Lightweight Directory Access Protocol (v3): Technical Specification*.
- IETF RFC 3403 (2002), *Dynamic Delegation Discovery System (DDDS) Part Three: The Domain Name System (DNS) Database*.
- IETF RFC 3513 (2003), *Internet Protocol Version 6 (IPv6) Addressing Architecture*.
- IETF RFC 3546 (2003), *Transport Layer Security (TLS) Extensions*.
- IETF RFC 3923 (2004), *End-to-End Signing and Object Encryption for the Extensible Messaging and Presence Protocol (XMPP)*.
- IETF RFC 4122 (2005), *A Universally Unique IDentifier (UUID) URN Namespace*.
- Liberty Alliance POAS:2003, R. Aarts, *Liberty Reverse HTTP Binding for SOAP Specification Version 1.0*, *Liberty Alliance Project*.
- OASIS WSS:2006, WS-Security Core Specification 1.1.
- UNICODE-C, M. Davis; M. J. Dürst: *Unicode Normalization Forms.* UNICODE Consortium, March 2001.
- W3C Canonicalization:2002, *Exclusive XML Canonicalization Version 1.0,* W3C Recommendation, Copyright © [18 July 2002] World Wide Web Consortium, (Massachusetts Institute of Technology, Institut National de Recherche en Informatique et en Automatique, Keio University), http://www.w3.org/TR/xml-exc-c14n/.

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

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

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

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

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

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

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

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

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

– W3C String:1998, *Requirements for String Identity Matching and String Indexing,* W3C Note, Copyright © [10 July 1998] World Wide Web Consortium, (Massachusetts Institute of Technology, Institut National de Recherche en Informatique et en Automatique, Keio University, http://www.w3.org/TR/WD-charreq.

– W3C SOAP:2000, *Simple Object Access Protocol (SOAP) 1.1,* W3C Note, Copyright © [08 May 2000] World Wide Web Consortium, (Massachusetts Institute of Technology, Institut National de Recherche en Informatique et en Automatique, Keio University), http://www.w3.org/TR/2000/NOTE-SOAP-20000508.

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

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

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

ПРИМЕЧАНИЕ. – Ссылка в настоящей Рекомендации на какой-либо документ не придает этому отдельному документу статуса Рекомендации.

## 3 Определения

В настоящей Рекомендации используются следующие определения.

### 3.1 Заимствованные определения

**3.1.1** В настоящей Рекомендации используются следующий термин, определенный в Рекомендации МСЭ-Т X.667:

a) UUID – универсальный уникальный идентификатор.

**3.1.2** В настоящей Рекомендации используются следующие термины, определенные в Рекомендации МСЭ-Т X.680:

a) Идентификатор объекта;

b) Нотация открытого типа.

**3.1.3** В настоящей Рекомендации используются следующий термин, определенный в Рекомендации МСЭ-Т X.811:

a) Принцип.

**3.1.4** В настоящей Рекомендации используются следующие термины, определенные в Рекомендации МСЭ-Т X.812:

a) Информация о контроле доступа;

b) Пользователь.

**3.1.5** В настоящей Рекомендации используются следующие термины, определенные в Словаре веб-услуг Консорциума W3C:

a) Исходный отправитель SOAP;

b) Область имен;

c) Окончательный приемник SOAP;

d) Схема XML.

**3.1.6** В настоящей Рекомендации используются следующие термины, определенные в IETF RFC 2828:

a) Доступ;

b) Контроль доступа;

c) Прокси;

d) Прокси сервер;

f) Модель приема сообщений с опросом;

e) Пассивная модель приема сообщений;

g) Архитектура безопасности;

h) Политика безопасности;

i) Служба безопасности.

**3.1.7** В настоящей Рекомендации используются следующие термины, определенные в IETF RFC 2396:

a) Унифицированный идентификатор ресурса (URI);

b) Ссылка URI.

### 3.2 Дополнительные определения

**3.2.1 права доступа (access rights)**: Описание типа разрешенных взаимодействий, который может выполнить объект при помощи денного ресурса. Например, читать, писать, исполнять, добавлять, изменять и удалять.

**3.2.2 аккаунт (account)**: Формальное соглашение по регулярному предоставлению услуг и выполнения оплаты между пользователем услуги и компанией – провайдером услуг.

**3.2.3 объединение аккаунтов (account linkage)**: Метод соединения аккауантов одного и того же клиента у двух различных провайдеров, так что эти провайдеры могут иметь данные об этом клиенте. Объединение аккаунтов может быть установлено при помощи совместного использования атрибутов или при помощи объединения идентификационной информации.

**3.2.4 активная роль (active role)**: Роль, которая назначена объекту системы для выполнения некоторых операций, например, доступа к ресурсам.

**3.2.5 административный домен (administrative domain)**: Условия или контекст, который определяется некоторой комбинацией одного или нескольких административных правил, Регистрация доменного имени в интернете, юридические лица (например, частные предприниматели, корпорации или другие формальные организации), плюс множество хостов, сетевых устройств и соединяющие их сети (и, возможно, другие параметры), плюс услуги сети (зачастую чрезвычайно разнообразные) и работающие с ними приложения.

Административный домен может содержать или определять один или несколько доменов безопасности. Административный домен охватывать один или несколько сайтов. Параметры, определяющие административный домен, могут и во многих случаях будут эволюционировать со временем. Административные домены могут взаимодействовать и заключать соглашения по предоставлению и/или потреблению услуг за пределами границ самого административного домена.

**3.2.6 администратор (administrator)**: Человек, который устанавливает и обслуживает систему, или который использует ее для управления элементами системы, пользователями и/или контентом. Администратор, как правило, принадлежит конкретному административному домену и может принадлежать нескольким административным доменам.

**3.2.7 принадлежность (affiliation); принадлежность к группе (affiliation group)**: Набор системных объектов, которые имеют общую область имен (в смысле объединения) идентификаторов для клиентов.

**3.2.8 анонимность (anonymity)**: Качество или состояние анонимности, которое является условием владения именем или идентификатора, которые неизвестны или замаскированы.

**3.2.9 подтверждающая сторона (asserting party)**: Формально, административный домен, в котором находится один или несколько органов управления SAML. Неформально, один из органов управления SAML.

**3.2.10 подтверждение (assertion)**: Блок данных, создаваемый органом управления SAML либо в ходе выполненного действия по аутентификации объекта, либо относительно данных аутентификации, применяемых к объекту в отношении определяемого ресурса.

**3.2.11 атрибут (attribute)**: Специфические характеристики объекта. Для объектов реального мира атрибуты часто указываются в форме физических параметров, таких как размер, форма, вес и цвет. В киберпространстве объекты могут иметь атрибуты, описывающие размер, тип кодирования, сетевой адрес и т. д. Атрибуты часто представлены в виде пар "имя атрибуты" и "значение(я) атрибута", например, атрибут "foo" имеет значение 'bar', "count" имеет значение 1, "gizmo" имеет значения "frob" и "2".

**3.2.12 подтверждение атрибута (attribute assertion)**: Подтверждение, которое переносит информацию об атрибутах объекта.

**3.2.13 ответственный орган проверки атрибута (attribute authority)**: Элемент системы, создающий подтверждения атрибута.

**3.2.14 аутентификация (authentication)**: Аутентификация – это процесс определения того, является ли на самом деле кто-то или что-то именно тем, кем он себя объявляет с данной степенью достоверности.

**3.2.15 подтверждение аутентификации (authentication assertion)**: Подтверждение, которое переносит информацию об успешном завершении процедуры аутентификации для данного объекта.

**3.2.16 ответственный орган аутентификации (authentication authority)**: Элемент системы, создающий подтверждения аутентификации.

**3.2.17 авторизация (authorization)**: Процесс определения, путем оценки соответствующей информации о контроле доступа, разрешено ли данному объекту иметь определенный тип доступа к конкретному ресурсу. Как правило, авторизация – это одна из частей аутентификации. После того как объект аутентифицирован, ему могут быть разрешены различные типы доступа.

**3.2.18 решение об авторизации (authorization decision)**: Результат действий по авторизации. Этот результат может быть отрицательным, т. е. он может показывать, что объекту не разрешен доступ к конкретному ресурсу.

**3.2.19 подтверждение решения об авторизации (authorization decision assertion)**: Подтверждение, которое переносит информацию решения об авторизации.

**3.2.20 обратный канал (back channel)**: Обратный канал обозначает непосредственную связь между двумя элементами системы без "перенаправления" сообщений через другой элемент системы, такой как, клиент HTTP (например, агент пользователя).

**3.2.21 связь (binding); связи протокола (protocol binding)**: В общем случае, спецификация преобразования особым образом сообщений некоторого данного протокола и, возможно, правила обмена сообщениями, в другой протокол. Например, преобразование сообщения языка SAML `<AuthnRequest>` в сообщение протокола HTTP является одним из примеров связи. Преобразование того же самого сообщения SAML в сообщение SOAP – это другая связь. В рамках языка SAML каждой связи присваивается имя виде "SAML xxx связь".

**3.2.22 данные для установления подлинности личности (credentials)**: Данные, которые передаются для установления идентификации заявляющего клиента.

**3.2.23 конечный пользователь (end user)**: Человек, который использует ресурсы для прикладных целей.

**3.2.24 элемент (entity)**: См. элемент системы.

**3.2.25 объединять (federate)**: Связывать или соединять вместе два или несколько объектов.

**3.2.26 объединение (federation)**: Этот термин используется в двух смыслах:

1) действие по установлению взаимосвязи между двумя элементами;
2) объединение, состоящее из любого количества провайдеров услуг и провайдеров идентификаторов.

**3.2.27 объединенная идентификация (federated identity)**: Говорят, что идентификация клиента должна быть объединенной для множества провайдеров, когда между провайдерами имеется соглашение о комплексе идентификаторов и/или атрибутов, которые должны использоваться для определения клиента.

**3.2.28 прямой канал (front channel)**: Прямым каналом называется "канал связи", который может быть установлен между двумя серверами, "разговаривающими" на языке HTTP, за счет применения сообщений "перенаправление HTTP" и передачи, таким образом, сообщений друг другу при помощи агента пользователя, например, веб-браузера или любого другого клиента HTTP.

**3.2.29 идентификатор (identifier)**: Объект данных (например, строка), преобразованный в элемент системы, который однозначно обозначает этот элемент системы. Элемент системы может иметь несколько различных идентификаторов, обозначающих его. Идентификатор главным образом является "отличительной чертой" элемента.

**3.2.30 идентификация (identity)**: Сущность объекта. Идентификация объекта часто описывается характеристиками объекта, среди которых может быть любое количество идентификаторов.

**3.2.31 разъединение идентификации (identity defederation)**: Действие, выполняемое, когда провайдеры соглашаются прекратить определение клиента при помощи определенного множества идентификаторов и/или атрибутов.

**3.2.32 объединение идентификации (identity federation)**: Действие по созданию объединенной идентификации от лица клиента.

**3.2.33 провайдер идентификации (identity provider)**: Тип провайдера услуг, который создает, поддерживает и управляет информацией об идентификации клиентов и обеспечивает аутентификацию клиентов для других провайдеров услуг внутри объединения, например профилей веб-браузеров.

**3.2.34 содержание провайдера идентификации (identity provider lite)**: Тип провайдера услуг, который создает, поддерживает и управляет информацией об идентификации клиентов и обеспечивает аутентификацию клиентов для других провайдеров услуг внутри объединения, используя требуемые участки языка SAML.

**3.2.35 вход в систему, начало сеанса, регистрация (login, logon, sign-on)**: Процесс, в ходе которого пользователь предоставляет органу по аутентификации данные для установления подлинности его личности, устанавливает сеанс связи и, возможно, устанавливает сеанс конференц-связи.

**3.2.36 выход из системы, завершение сеанса, отмена регистрации (logout, logoff, sign-off)**: Процесс, в ходе которого пользователь выражает желание завершить простой сеанс связи или сеанс конференц-связи.

**3.2.37 язык разметки (markup language)**: Набор элементов XML и атрибутов XML, которые должны быть использованы для конкретной цели в структуре документа XML. Язык разметки, как правило, определяется при помощи множества схем XML и сопроводительной документации к ним.

**3.2.38 описатель имени (name qualifier)**: Строка, которая устраняет неоднозначность идентификатора, который может быть использован в нескольких областях имен (в смысле объединения) для представления различных клиентов.

**3.2.39 сторона (party)**: Неформально – один или несколько клиентов, участвующих в некотором процессе или в обмене информацией, например, передачи подтверждения или оценки доступности ресурса.

**3.2.40 постоянный псевдоним (persistent pseudonym)**: Идентификатор имени, обеспечивающие его секретность, назначаемый провайдером с целью идентификации клиента для данной доверяющей стороны на протяжении длительного периода времени, который охватывает несколько сеансов связи; может использоваться для представления объединения идентификаторов.

**3.2.41 точка принятия стратегического решения (policy decision point (PDP))**: Элемент системы, который принимает решения об авторизации для самого себя или для других элементов системы, которые запрашивают такие решения. Например, PDP языка SAML получает запросы на решение по авторизации и формирует в ответ подтверждения решений об авторизации. PDP – это "орган, принимающий решения об авторизации ".

**3.2.42 точка обязательного выполнения стратегического решения (policy enforcement point (PEP))**: Элемент системы, который запрашивает и затем обеспечивает выполнение решений об авторизации. Например, PEP языка SAML передает запросы на решение по авторизации на PDP и получает подтверждения решений об авторизации, переданные в ответ на эти запросы.

**3.2.43 идентификация клиента (principal identity)**: Представление идентификации клиента, обычно это – идентификатор.

**3.2.44 профиль (profile)**: Множество правил для одной из нескольких целей; каждому множеству присваивается имя в виде "xxx профиль SAML" или "xxx SAML профиль":

1) Правила относительно того, как вводить подтверждения в протокол и извлекать их из протокола или другого используемого контекста.
2) Правила использования протокольных сообщений SAML в определенных условиях применения.
3) Правила преобразования атрибутов, выраженных на языке SAML в другую систему представления атрибутов. Этот набор правил известен как "профиль атрибута".

**3.2.45 связи протокола (protocol binding)**: См. "Binding".

**3.2.46 провайдер (provider)**: Общее название для провайдеров идентификации и для провайдеров услуг.

**3.2.47 доверяющая сторона (relying party)**: Элемент системы, который принимает решение о выполнении действий на основе информации от элемента другой системы. Например, доверяющая сторона SAML зависит от принимаемых подтверждений об объекте от подтверждающей стороны (ответственного органа SAML).

**3.2.48 запрашивающая сторона (requester)**: Элемент системы, который использует протокол SAML для запроса услуг от другого элемента системы (ответственного органа SAML, отвечающей стороны). В этой нотации термин "клиент" не используется, поскольку многие элементы системы одновременно или последовательно действуют как клиенты и как серверы. В тех случаях, когда используется связь SOAP для языка SAML, запрашивающая сторона SAML архитектурно отделена от исходного передатчика SOAP.

**3.2.49 ресурс (resource)**: Данные, содержащиеся в информационной системе (например, в виде файлов, информации в памяти и т. д.), а также:

1) услуга, предоставляемая системой;
2) единица оборудования системы (другими словами – компонент системы, например, аппаратура, стандарты, программное обеспечение или документация).

**3.2.50 отвечающая сторона (responder)**: Элемент системы (ответственный орган SAML), который использует протокол SAML для ответов на запросы услуг от другого элемента системы (запрашивающей стороны). В этой нотации термин "сервер" используется, поскольку многие элементы системы одновременно или последовательно действуют как клиенты и как серверы. В тех случаях, когда используется связь SOAP для языка SAML, отвечающая сторона SAML архитектурно отделена от окончательного приемника SOAP.

**3.2.51 роль (role)**: Словари определят слово "роль", как "персонаж или герой, исполняемый актером", либо "функция или позиция". Элементы системы выполняют различные роли последовательно и/или одновременно, например, активные роли и пассивные роли. Нотация "администратор" часто является примером роли.

**3.2.52 артефакт SAML (SAML artifact)**: Небольшой структурированный объект данных фиксированного размера, указывающий на, как правило, много большее протокольное сообщение SAML с переменным размером. Артефакты SAML разработаны так, чтобы их можно было вводить в URL и передавать в сообщениях протокола HTTP, например, ответных сообщениях HTTP с кодами статуса "3xx Redirection", и последующих сообщениях GET протокола HTTP. Таким образом, провайдер услуг может опосредованно через агента пользователя передавать артефакт SAML другому провайдеру, который затем может отменить указание на артефакт SAML при помощи прямого взаимодействия с провайдером, предоставляющим эту услугу, и получить сообщение протокола SAML.

**3.2.53 ответственный орган SAML (SAML authority)**: Объект абстрактной системы в модели домена SAML, который формирует подтверждения. См. также ответственный орган проверки атрибута, ответственный орган аутентификации и точка принятия стратегического решения (PDP).

**3.2.54 безопасность (security)**: Комплекс охранных мер, которые обеспечивают конфиденциальность информации, защищают системы или сети, используемые для ее обработки, и регулирует доступ к ним. Безопасность, как правило, охватывает аспекты секретности, конфиденциальности, целостности и доступности. Ее назначение обеспечить сопротивление системы возможно коррелированным атакам.

**3.2.55 подтверждение безопасности (security assertion)**: Подтверждение, которое тщательно проверяется в контексте архитектура безопасности.

**3.2.56 контекст безопасности (security context)**: По отношению к одному сообщению протокола SAML, контекст безопасности этого сообщения представляет собой семантическое объединение элементов заголовка безопасности сообщения (если они имеются) с другими механизмами безопасности, которые могут использоваться при доставке сообщения получателю. По отношению к последнему, примером являются механизмы безопасности, используемые на низших уровнях сетевого стека, таких как HTTP, TLS и IPSec.

**3.2.57 домен безопасность (security domain)**: Условия или контекст, который определяется моделями безопасности и архитектурой безопасности, включая комплекс ресурсов и множество элементов системы, которым разрешен доступ к ресурсам. В одном административном домене может располагаться один или несколько доменов безопасность. Параметры, определяющие данный домен безопасности, как правило, эволюционируют со временем.

**3.2.58 выражение политики безопасности (security policy expression)**: Определение соответствия идентификаторов и/или атрибутов клиентов и разрешенных им действий. Выражением политики безопасности часто являются списки контроля доступа.

**3.2.59 service provider (провайдер услуг)**: Роль, выполняемая элементом системы, когда элемент системы предоставляет услуги клиентам и/или другим элементам системы.

**3.2.60 содержание провайдера услуг (service provider lite)**: Роль, выполняемая элементом системы, когда элемент системы предоставляет услуги клиентам и/или другим элементам системы, используя только требуемую часть протокола SAML.

**3.2.61 сеанс связи (session)**: Продолжительное взаимодействие между элементами системы, в котором часто участвует клиент, характеризующееся поддержанием определенного типа взаимодействия на всем протяжении этого взаимодействия.

**3.2.62 администратор сеанса связи (session authority)**: Роль, выполняемая элементом системы, когда он поддерживает состояние, соответствующее сеансам связи.

**3.2.63 участник сеанса связи (session participant)**: Роль, выполняемая элементом системы, когда он участвует в сеансе связи как минимум с администратором сеанса связи.

**3.2.64 отказаться от регистрации (sign-off)**: См. "logout".

**3.2.65 зарегистрироваться (sign-on)**: См. "login".

**3.2.66 сайт (site)**: Неформальный термин, описывающий административный домен его географическим именем или именем DNS. Он может определять конкретный географический или топографический участок административного домена, или может охватывать несколько административных доменов, как в случае сайта ASP.

**3.2.67 субъект (subject)**: Клиент в контексте домена безопасности. Подтверждения SAML заявляют о субъектах.

**3.2.68 элемент системы, элемент (system entity; entity)**: Активный компонент компьютерной системы/сети. Например, автоматизированный процесс или множество процессов, подсистема, человек или группа людей, которые обладают отличительным набором функций.

**3.2.69 период ожидания (time-out)**: Промежуток времени, после завершения которого некоторые условия выполняются, если не происходит определенного события. Например, сеанс связи завершается, поскольку его состояние было не активным в течение определенного периода времени, называемого периодом ожидания.

**3.2.70 временный псевдоним (transient pseudonym)**: Идентификатор имени, обеспечивающие его секретность, назначаемый провайдером с целью идентификации клиента для данной доверяющей стороны на относительно небольшой период времени, который не охватывает несколько сеансов связи.

**3.2.71 атрибут XML (XML attribute)**: Структура данных XML, которая вводится в стартовую команду элемента XML, и которая имеет название и значение.

**3.2.72 элемент XML (XML element)**: Структура данных XML, которая располагается в иерархическом порядке среди других таких структур в документе XML и выделяется либо стартовой командой, либо пустой командой

## 4 Сокращения

В настоящей Рекомендации используются следующие сокращения.

| | | |
|---|---|---|
| AA | Attribute Authority | Ответственный орган проверки атрибута |
| ASN.1 | Abstract Syntax Notation One | Абстрактная синтаксическая нотация 1 |
| ASP | Application Service Provider | Провайдер прикладных услуг |
| CA | Certification Authority | Ответственный орган по сертификации |
| CMP | Certificate Management Protocol | Протокол управления сертификатами |
| CRL | Certificate Revocation List | Список отозванных сертификатов |
| DCE | Distributed Computing Environment | Среда распределенных вычислений |
| DDDS | Dynamic Delegation Discovery System | Система динамического обнаружения ресурсов |
| DNS | Domain Name System | Система доменных имен |
| ECP | Enhanced Client/Proxy | Расширенная архитектура "клиент/прокси-сервер" |
| HTTP | HyperText Transfer Protocol | Протокол передачи гипертекста |
| HTTPS | Secure HyperText Transfer Protocol | Защищенный протокол передачи гипертекста |
| IdP | Identity Provider | Провайдер идентификации |
| IdP Lite | Identity Provider Lite | Содержание провайдера идентификации |
| IP | Internet Protocol | Протокол Интернет |
| IPSec | Internet Protocol Security | Безопасный протокол Интернет |
| MD5 | Message Digest algorithm 5 | Алгоритм хэширования (преобразования входного массива данных в короткое число фиксированной длины) сообщений 5 |
| MIME | Multipurpose Internet Mail Extensions | Многоцелевые расширения электронной почты в интернете |
| NAPTR | Naming Authority PoinTeR | Указатель владельца имени |
| OID | Object IDentifier | Идентификатор объекта |
| PAC | Privilege Attribute Certificates | Сертификат атрибута привилегий |
| PAOS | Reverse SOAP | Обратный простой протокол доступа к объекту |
| PDP | Policy Decision Point | Точка принятия стратегического решения |
| PEP | Policy Enforcement Point | Точка обязательного выполнения стратегического решения |
| PGP | Pretty Good Privacy | Надежная конфиденциальность, Программа для шифрования сообщений электронной почты. Основана на алгоритме шифрования с открытым ключом |
| PKI | Public-Key Infrastructure | Инфраструктура открытого ключа |
| POP | Proof of Possession | [протокол] доказательства владения |
| RA | Registration Authority | Ответственный орган регистрации |
| RSA | Rivest, Shamir, Adleman (public key algorithm) | Алгоритм шифрования с открытым ключом. Название алгоритма образовано из первых букв фамилий его авторов: Ron Rivest, Adi Shamir, Leonard Adleman |

| | | |
|---|---|---|
| SHA-1 | Secure Hash Algorithm 1 | Защищенный алгоритм хеширования 1 |
| SP | Service Provider | Провайдер услуг |
| SPKI | Simple Public Key Infrastructure | Простая инфраструктура открытого ключа |
| SP Lite | Service Provider Lite | Содержание провайдера услуг |
| SSO | Single Sign On | Единая регистрация во всей сети путем однократного ввода пароля |
| TLS | Transport Layer Security protocol | Протокол безопасности на транспортном уровне |
| URI | Uniform Resource Identifier | Унифицированный идентификатор ресурса |
| UTC | Coordinated Universal Time | Универсальное скоординированное время |
| UUID | Universal Unique IDentifier | Универсальный уникальный идентификатор |
| XACML | eXtensible Access Control Markup Language | Расширяемый язык разметки контроля доступа |
| XML | eXtensible Markup Language | Расширяемый язык разметки |

## 5 Условные обозначения

В настоящей Рекомендации используются слова "должен", "нельзя", "требуемый", "следует", "не следует", "обязан", "не должен", "рекомендованный", "может" и "дополнительный". В настоящей Рекомендации эти термины должны интерпретироваться так, как это описано в документе IETF RFC 2119.

В настоящей Рекомендации используются документы схемы XML, соответствующие Схеме XML Консорциума W3C – Часть 1, Схеме XML Консорциума W3C – Часть 2 и нормативным текстам этих спецификаций в том, что касается описания синтаксиса и семантики Подтверждений SAML в кодировке XML и протокольных сообщений. В случаях расхождений между документами Схемы SAML и листингами схемы, приведенными в настоящей Рекомендации, предпочтение следует отдавать документам схемы. Отметим, что в некоторых случаях настоящая Рекомендация налагает ограничения более строгие, чем те, что определены документами схемы.

## 6 Обзор

Настоящая Рекомендация определяет спецификацию версии 2 Языка разметки, предусматривающего защиту данных (SAML v2.0). Она определяет синтаксис и семантику обработки подтверждений, сделанных элементом системы относительно объекта. В процессе формулирования или применения таких подтверждений, Элементы системы SAML могут использовать другие протоколы для связи либо с самим подтверждением, либо с объектом подтверждения. В настоящей Рекомендации, в дополнение к правилам обработки, используемым при управлении системой SAML, определяется структура Подтверждений языка SAML и связанный с ними набор протоколов.

Подтверждения SAML и протокольные сообщения кодируются в виде XML и используют области имен XML. Как правило, для транспортировки они встраиваются в другие структуры, такие как запросы HTTP POST или SOAP сообщения в кодировке XML. В разделе 7 определяются общие типы данных, используемых в SAML. В разделе 8 приведены базовые принципы для Подтверждений и протоколов SAML. В разделе 9 описывается модель метаданных SAML. В разделе 10 разрабатываются принципы введения и транспортировки протокольных сообщений языка SAML. В разделе 11 приведен базовый набор профилей для использования Подтверждений и протоколов SAML с целью выполнения конкретных задач или обеспечения взаимодействия при использовании возможностей SAML. В разделе 12 рассматриваются аспекты аутентификации для языка SAML. В частности, определяются следующие аспекты:

- Схема SAML – Правила аутентификации;
- Схема SAML – Правила аутентификации типов;
- Схема класса контекста SAML для протокола Интернет;
- Схема класса контекста SAML для пароля протокола Интернет;
- Схема класса контекста SAML для службы сетевой аутентификации Kerberos;
- Схема класса контекста SAML для однофакторного незарегистрированного мобильного абонента;
- Схема класса контекста SAML для двухфакторного незарегистрированного мобильного абонента;
- Схема класса контекста SAML для однофакторного контракта мобильного абонента;
- Схема класса контекста SAML для двухфакторного контракта мобильного абонента;
- Схема класса контекста SAML для пароля;
- Схема класса контекста SAML для транспортировки с парольной защитой;
- Схема класса контекста SAML для предыдущего сеанса связи;
- Схема класса контекста SAML для открытого ключа – X.509;
- Схема класса контекста SAML для открытого ключа – PGP;

- Схема класса контекста SAML для открытого ключа – SPKI;
- Схема класса контекста SAML для открытого ключа – подпись XML;
- Схема класса контекста SAML для смарт-карты;
- Схема класса контекста SAML для PKI со смарт-картой;
- Схема класса контекста SAML для PKI с программной защитой;
- Схема класса контекста SAML для телефонии;
- Схема класса контекста SAML для телефонии (номадической);
- Схема класса контекста SAML для телефонии (персональной);
- Схема класса контекста SAML для телефонии (аутентифицированной);
- Схема класса контекста SAML для безопасного удаленного доступа с паролем;
- Схема класса контекста SAML для аутентификации клиента на базе сертификатов SL/TLS;
- Схема класса контекста SAML для синхронных меток.

В разделе 13 приведены основные принципы для тех, кто будет реализовывать SAML, которые необходимо выполнять для обеспечения соответствия. В разделе 13 рассматриваются требования к соответствию, включая режимы работы и модели безопасности. Приложение A содержит перечень всех соответствующих схем SAML.

## 7 Общие типы данных

В последующих разделах определяется. Как использовать и интерпретировать общие типы данных, которые появляются в Схемах SAML.

### 7.1 Строчные значения

Все строчные значения SAML имеют тип **xs:string** (строка), который входит в состав типов данных W3C XML. Если в настоящей Рекомендации не указано иного, все строки в сообщениях SAML должны содержать как минимум один символ, не являющийся пробелом.

Если в настоящей Рекомендации или в конкретных профилях не указано иного, все элементы в документах SAML, которые имеют тип Схемы XML **xs:string**, или тип, полученный из этого типа, должны сравниваться с использованием точного побитового сравнения. В частности, варианты реализации и построения SAML не должны зависеть от сравнения строк без учета регистра, нормализации или вычищения пробелов или преобразования в местные форматы типа нумерации или валюты. Это требование введено для обеспечения согласования с типом Строка W3C.

Если в некотором варианте сравниваются значения, выраженные с использованием различных сем кодировки символов, в этом варианте следует применять метод сравнения, который выдает такой же результат, который был бы получен при преобразовании обоих величин в символы в кодировке Unicode, Normalization Form C и последующем выполнении точного побитового сравнения. Это требование введено для обеспечения согласования с Моделью символа W3C, в частности с правилами для текста в нормализованной кодировке Unicode.

Приложения, в которых сравниваются данные, полученные в виде документов SAML, с данными из внешних источников, должны учитывать правила нормализации, определенные для XML. Текст, содержащийся внутри элементов, нормализуется так, чтобы окончания строк были представлены символом перевода строки (ASCII CODE $10_{Decimal}$). Значения атрибутов XML, определенных как строки (или типы, полученные из строк), нормализуются как описано в документе W3C XML 1.0, 3.3.3. Все пробелы заменяются пустыми символами (ASCII CODE $32_{Decimal}$).

Настоящая Рекомендация не определяет объединение или порядок сортировки для значений атрибутов XML или содержания элементов. Варианты реализации SAML не должны зависеть от конкретных порядков сортировки для значений, поскольку они могут различаться в зависимости от местных установок на серверах, участвующих в процессе.

### 7.2 Значения Унифицированного идентификатора ресурса

Все эталонные значения URI языка SAML имеют тип **xs:anyURI** (любой URI), который входит в состав типов данных W3C XML.

Если в настоящей Рекомендации не указано иного, все эталонные значения URI, которые используются в элементах или атрибутах, определенных в языке SAML, должны состоять из как минимум одного символа, не являющегося пробелом, и требуется, чтобы эти значения были абсолютными.

В настоящей Рекомендации указатели URI широко используются в виде идентификаторов, таких как коды состояния, типы формата, названия атрибутов и элементов системы и т. д. Следовательно, очень важно, чтобы эти значения были бы и уникальными, и согласованными, так например, в разное время для описания различной базовой информации не может использоваться один и тот же URI.

### 7.3 Значения времени

Все значения времени SAML имеют тип **xs:dateTime** (дата и время), который входит в состав типов данных W3C XML, и должны быть представлены в форме UTC без компонентов временного пояса.

Элементы системы SAML не должны основываться на единицах времени менее миллисекунд. Варианты реализации не должны формировать временные интервалы, которые определяют корректировочные секунды.

### 7.4 Значения ID и ссылки на ID

Простой тип **xs:ID** используется для объявления идентификаторов SAML для подтверждений, запросов и ответов. Значения, для которых в настоящей Рекомендации определено, что они имеют тип **xs:ID,** должны обладать следующими свойствами в дополнение к свойствам, которые должен иметь по определению сам тип **xs:ID**:

- любая сторона, которая назначает идентификатор, должна гарантировать, что вероятность случайного назначения третьей стороной или любой иной стороной этого же идентификатора другому объекту данных пренебрежимо мала;
- если объект данных объявляет о том, что он имеет идентификатор, должно существовать только одно такое заявление.

Механизм, при помощи которого элемент системы SAML обеспечивает уникальность идентификатора, зависит от варианта реализации. В случае применения метода случайных или псевдослучайных последовательностей вероятность того, что два случайно выбранных идентификатора окажутся идентичными, должна бать меньше или равна $2^{-128}$ и желательно, чтобы она была меньше или равна $2^{-160}$. Это требование может быть выполнено при помощи кодирования случайно выбранного значения длиной от 128 до 160 битов. Это кодирование должно соответствовать правилам, определенным типом данных **xs:ID**. Для того чтобы обеспечить желаемую уникальность параметров для различных систем, на вход псевдослучайного генератора должен быть подан уникальный материал.

Простой тип **xs:NCName** используется в языке SAML для обозначения идентификаторов типа **xs:ID,** поскольку тип **xs:IDREF** не может быть использован для этой цели. В языке SAML, элемент, обозначенный ссылкой на идентификатор SAML, может, на самом деле, быть определен в документе, отличном от того, где используется эта ссылка на идентификатор. Применение типа **xs:IDREF** могло бы нарушить требование, согласно которому его значение в одном и том же документе XML на некоторых элементах должно соответствовать значению атрибута ID.

## 8 Подтверждения и протоколы SAML

Язык SAML определяет синтаксис и семантику обработки подтверждений, сделанных элементом системы относительно объекта. В процессе формулирования или применения таких подтверждений, Элементы системы SAML могут использовать другие протоколы для передачи либо самого подтверждения, либо субъекта подтверждения. В настоящем разделе, в дополнение к правилам обработки, используемым при управлении системой SAML, определяется структура Подтверждений языка SAML и связанный с ними набор протоколов.

Подтверждения SAML и протокольные сообщения кодируются в виде XML (см. W3C XML 1.0) и используют области имен XML (см. Области имен W3C). Как правило, для транспортировки они встраиваются в другие структуры, такие как запросы HTTP POST или SOAP сообщения в кодировке XML. В разделе 10 описываются принципы введения и транспортировки протокольных сообщений языка SAML. В разделе 11 приведен базовый набор профилей для использования Подтверждений и протоколов SAML с целью выполнения конкретных задач или обеспечения взаимодействия при использовании возможностей SAML.

### 8.1 Подтверждения SAML

Подтверждение – это блок информации, который содержит ноль или несколько утверждений, сделанных **ответственным органом SAML**; ответственные органы SAML иногда называют **подтверждающими сторонами** в рассуждениях о формировании, передаче и приеме подтверждений, а элементы системы, которые используют принятые подтверждения, известны под названием, **доверяющей стороны**. (Эти термины отличаются от терминов **запрашивающая сторона** и **отвечающая сторона**, которые предназначены для обсуждения вопросов протокола обмена сообщениями SAML.)

Подтверждения SAML обычно делается относительно **субъекта,** представленного элементом <Subject>. Однако элемент <Subject> является дополнительным, а другие спецификации и профили могут использовать структуру подтверждений SAML для создания аналогичных утверждений без определения субъекта, или, возможно, определяя субъект иным способом. Как правило, существует множество **провайдеров услуг**. Которые могут использовать подтверждения для субъекта для контроля доступа и предоставления определенных услуг, и, соответственно, они становятся доверяющими сторонами для подтверждающей стороны, которая называется провайдером идентификации.

Настоящая Рекомендация определяет три различных вида утверждений о подтверждении, которые могут быть сформированы ответственным органом SAML. Все утверждения, определенные в языке SAML, связываются с объектом. В настоящей Рекомендации определены следующие три вида утверждений:

- **Authentication (Аутентификация)**: Субъект подтверждения был аутентифицирован определенными средствами в определенный момент времени.
- **Attribute (Атрибут)**: Субъект подтверждения ассоциируется с предоставленными атрибутами.
- **Authorization decision (Решение об авторизации)**: Просьба разрешить субъекту подтверждения доступ к определенным ресурсам была удовлетворена или отклонена.

ПРИМЕЧАНИЕ (информативное). – PE13 (см. OASIS PE:2006) предлагает добавить к вышеприведенному параграфу "или решение является неопределенным".

Внешняя структура подтверждения является общей, предоставляющей информацию, которая одинакова для всех утверждений внутри него. Внутри подтверждение множество внутренних элементов описывает аутентификацию, атрибут, решение об авторизации или утверждения пользователя, содержащие конкретные подробности.

Расширения разрешены схемой подтверждения языка SAML, как описано в 8.6, допускающей определенные пользователем расширения подтверждений и утверждений, а также допускающей определение новых типов подтверждений и утверждений.

### 8.1.1 Описания заголовка схемы и области имен

Приведенный далее фрагмент схемы определяет области имен XML и другую информацию заголовка для схемы подтверждения:

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

### 8.1.2 Идентификаторы имен

В последующих разделах определяются конструкции SAML, которые содержат описательные идентификаторы объектов и создателей подтверждений и протокольных сообщений.

В языке SAML существует множество ситуаций, когда необходимо, чтобы два элемента системы обсуждали третью сторону; например, протокол запроса аутентификации SAML позволяет третьей стороне аутентифицировать объект. Таким образом, полезно иметь средства, при помощи которых стороны могут быть ассоциированы с идентификаторами, которые имеют значение для каждой из сторон. В некоторых случаях, потребуется ограничить область применения идентификатора небольшим набором элементов системы (например, для сохранения секретности объекта). Аналогичные идентификаторы могут также использоваться для обозначения создателя протокольного сообщения SAML или подтверждения.

Вполне возможно, что два или несколько элементов системы могут использовать одно и то же значение идентификатора имени, когда ссылаются на различные идентификаторы. Таким образом, каждый элемент может иметь различное понимание одного и того же имени. Язык SAML предоставляет **спецификаторы имен** для устранения возможности различного толкования идентификатора имени, при помощи его эффективного расположения в объединенной **области имен** по отношению к спецификаторам имен. SAML v2.0 дает возможность определить идентификатор в понятиях, как подтверждающей стороны, так и конкретной доверяющей стороны или сообщества сторон, что позволяет идентификаторам, при необходимости, проявлять парную семантику.

Идентификаторы имен могут также шифроваться для еще большего улучшения параметров сохранения секретности, в частности, в тех случаях, когда идентификатор может передаваться через посредника.

ПРИМЕЧАНИЕ. – Во избежание использования относительно продвинутых конструкций схем XML, различные типы элементов идентификатора не используют общую иерархию типов.

#### 8.1.2.1 Элемент <BaseID>

Элемент <BaseID> – это точка расширения, которая позволяет приложениям добавлять новые типы идентификаторов. Его комплексный тип **BaseIDAbstractType** является абстрактным и, следовательно, применим только как основа для производного типа. Он содержит следующие атрибуты, которые используются расширенными представлениями идентификатора:

– NameQualifier [Дополнительный]

Домен безопасности или административный домен, который классифицирует идентификатор. Этот атрибут предоставляет средства для объединения идентификаторов из разделенных данных пользователей без коллизий.

– SPNameQualifier [Дополнительный]

Более точно определяет идентификатор, указывая имя провайдера услуг или сообщества провайдеров. Этот атрибут предоставляет дополнительные средства для объединения идентификаторов на основе доверяющей стороны или сторон.

Атрибуты NameQualifier и SPNameQualifier должны быть пропущены, если только определение типа идентификатора не определяет в явном виде их использование и семантику.

Приведенный далее фрагмент схемы определяет элемент <BaseID> и его сложный тип **BaseIDAbstractType**:

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

#### 8.1.2.2 Сложный тип NameIDType

Сложный тип **NameIDType** используется, когда элемент данных применяется для представления элемента в виде строчного имени. Это – более ограниченная форма идентификатора, чем элемент <BaseID> и является типом, лежащим в основе и элемента <NameID> и элемента <Issuer>. В дополнение к контенту строки, содержащей реальный идентификатор, он содержит следующие дополнительные атрибуты:

– NameQualifier [Дополнительный]

Домен безопасности или административный домен, который классифицирует имя. Этот атрибут предоставляет средства для объединения имен из разделенных данных пользователей без коллизий.

– SPNameQualifier [Дополнительный]

Более точно определяет имя, указывая имя провайдера услуг или сообщества провайдеров. Этот атрибут предоставляет дополнительные средства для объединения имен на основе доверяющей стороны или сторон.

– Format [Дополнительный]

Ссылка URI, представляющая классификацию информации строчных идентификаторов. В разделе 8.7.3 описаны ссылки URI, определенные в языке SAML, которые могут использоваться как значения атрибута Format, связанные с ними описания и правила обработки. Если элементом, основанным на этом типе, не установлено иное, то, если значение атрибута Format не указано, то используется urn:oasis:names:tc:SAML:1.0:nameid-format:unspecified (см. 8.7.3.1).

Когда используется значение Format, отличное от того, которое определено в 8.7.3, содержание элемента этого типа должно интерпретироваться в соответствии с определением данного формата, как определено вне рамок настоящей Рекомендации. Если определением данного формата не установлено иное, аспекты анонимности, назначения псевдонимов и предоставления идентификатора для подтверждающей и доверяющей стороны зависит от варианта реализации.

– SPProvidedID [Дополнительный]

А идентификатор имени, установленный для данного элемента провайдером услуг или сообществом провайдеров, отличается от первичного идентификатора имени, данного в содержании элемента. Этот атрибут предоставляет средства совместного использования идентификаторов SAML с существующими идентификаторами, которые уже используются провайдером услуг. Например, существующий идентификатор может быть "присоединен" к элементу с помощью протокола управления идентификаторами имен, определенного в 8.2.8.

Дополнительные правила по включению в содержание (или пропуска) этих атрибутов могут быть определены элементами, которые должны будут использовать этот тип, и определениями конкретного атрибута Format. Атрибуты NameQualifier и SPNameQualifier должны быть пропущены, если только элемент или формат не определяет в явном виде их использование и семантику.

Приведенный далее фрагмент схемы определяет сложный тип **NameIDType**:

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

#### 8.1.2.3 Элемент <NameID>

Элемент `<NameID>` имеет тип **NameIDType** (см. 8.1.2.2) и используется в различных конструкциях подтверждений SAML, таких как элементы `<Subject>` и `<SubjectConfirmation>`, и в различных протокольных сообщениях (см. 8.2).

Приведенный далее фрагмент схемы определяет элемент `<NameID>`:

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

#### 8.1.2.4 Элемент <EncryptedID>

Элемент `<EncryptedID>` имеет тип **EncryptedElementType**, он в зашифрованном виде переносит содержание нешифрованного элемента, как определено Правилами шифрования W3C. Элемент `<EncryptedID>` содержит следующие элементы:

– `<xenc:EncryptedData>` [Требуемый]

  Зашифрованное содержание и соответствующие данные шифрования определяются Правилами шифрования W3C. Желательно, чтобы был представлен атрибут "Тип" (Type) и, если он представлен, он должен содержать значение `http://www.w3.org/2001/04/xmlenc#Element`. Зашифрованное содержание должно содержать элемент типа **NameIDType** или типа **AssertionType**, или типа, который является производным от типа **BaseIDAbstractType**, **NameIDType** или **AssertionType**.

– `<xenc:EncryptedKey>` [Ноль или несколько]

  Свернутые ключи шифрования, как определено Правилами шифрования W3C. Каждый свернутый ключ должен содержать атрибут "Получатель" (Recipient), который определяет элемент, для которого был зашифрован ключ. Значением атрибута Получатель должен быть идентификатор URI элемента системы SAML, как определено в 8.4.

Зашифрованные идентификаторы выполняют функции механизма обеспечения секретности, когда обычные текстовые сообщения передаются через посредника. Таким образом, зашифрованный текст должен быть уникальным для любой отдельно взятой операции шифрования. Более подробно эти вопросы рассмотрены в Правилах шифрования W3C XML, 6.3.

В этом элементе может быть зашифровано полное подтверждение, и использоваться как идентификатор. В таком случае элемент `<Subject>` зашифрованного подтверждения представляет "идентификатор" объекта ограничивающего подтверждения. Отсюда, если подтверждение идентификации недействительно, то это подтверждение является ограничивающим.

Приведенный далее фрагмент схемы определяет элемент `<EncryptedID>` и его сложный тип **EncryptedElementType**:

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

#### 8.1.2.5 Элемент <Issuer>

Элемент `<Issuer>`, имеющий сложный тип **NameIDType**, содержит информацию о создателе подтверждения SAML или протокольного сообщения. Элемент требует использования строки для передачи имени создателя, но позволяет применять различные блоки описательной информации (см. 8.1.2.2).

В отличие от обычного правило для данного типа элемента, если в этом элементе не содержится значения `Format`, используется значение `urn:oasis:names:tc:SAML:2.0:nameid-format:entity` (см. 8.1.2.2).

Приведенный далее фрагмент схемы определяет элемент <Issuer>:

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

### 8.1.3 Подтверждения

В последующих разделах определяются конструкции языка SAML, которые либо содержат информацию подтверждения, либо предоставляют средства для ссылок на существующие подтверждения.

#### 8.1.3.1 Элемент <AssertionIDRef>

Элемент <AssertionIDRef> ссылается на подтверждение SAML, указывая его уникальный идентификатор. Конкретный ответственный орган, который создает подтверждение, или от которого это подтверждение может быть получено, не указывается как часть ссылки. Элемент протокола, который использует эту ссылку для запроса соответствующего подтверждения, описан в 8.2.3.

Приведенный далее фрагмент схемы определяет элемент <AssertionIDRef>:

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

#### 8.1.3.2 Элемент <AssertionURIRef>

Элемент <AssertionURIRef> ссылается на подтверждение SAML, указывая ссылку на URI. Эта ссылка на URI может использоваться для получения соответствующего подтверждения тем способом, который присущ данной ссылке на URI. Сведения о том, как этот элемент используется в протокольных связях для выполнения этой задачи, приведены в 7.3.

Приведенный далее фрагмент схемы определяет элемент <AssertionURIRef>:

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

#### 8.1.3.3 Элемент <Assertion>

Элемент <Assertion> имеет сложный тип **AssertionType**. Этот тип определяет базовую информацию, которая является общей для всех подтверждений, включая следующие элементы и атрибуты:

– Version [Требуемый]

  Версия этого подтверждения. Идентификатор для Версии языка SAML, определенной в настоящей Рекомендации, имеет значение "2.0". Контроль версий SAML рассматривается в 8.3.

– ID [Требуемый]

  Идентификатор для этого подтверждения. Он имеет тип **xs:ID**, и должен отвечать требованиям, определенным в 7.3 для уникальности идентификатора.

– IssueInstant [Требуемый]

  Момент времени создания, выраженный в UTC, как описано в 7.3.

– <Issuer> [Требуемый]

  Ответственный орган SAML, который создает запрос(ы) на подтверждение. Создатель должен быть однозначно определен для соответствующих доверяющих сторон.

  Настоящая Рекомендация не определяет специального взаимодействия между элементом системы, представляемым этим элементом и тем, кто подписывает подтверждение (если таковые есть). Любые требования, предъявляемые доверяющей стороной, которая использует подтверждение, или конкретными профилями, определяются приложением.

– <ds:Signature> [Дополнительный]

  Подпись XML, которая защищает целостность подтверждения и аутентифицирует его создателя, как описано далее и в 8.4.

– <Subject> [Дополнительный]

  Объект утверждения(й) в подтверждении.

– <Conditions> [Дополнительный]

  Условия, которые должны быть учтены при оценке применимости подтверждения и/или при использовании этого подтверждения. Дополнительная информация о том, как оценивать условия, приведена в 8.1.5.

– `<Advice>` [Дополнительный]

  Дополнительная информация, относящаяся к подтверждению и способствующая его обработке в определенных условиях, но которая может быть проигнорирована приложениями, которые не понимают совета или не желают им воспользоваться.

  Ноль или несколько следующих элементов утверждения:

– `<Statement>`

  Утверждение, тип которого определен в схеме расширения. Для указания используемого типа утверждения должен использоваться атрибут **xsi:type**.

– `<AuthnStatement>`

  Утверждение об аутентификации.

– `<AuthzDecisionStatement>`

  Утверждение о принятии решения об авторизации.

– `<AttributeStatement>`

  Утверждение об атрибуте.

Подтверждение без утверждений должно содержать элемент `<Subject>`. такое подтверждение идентифицирует клиента, который может быть назван или подтвержден методами SAML, но не подтверждает никакой дополнительной информации об этом клиенте.

В ином случае элемент `<Subject>`, если он представлен, идентифицирует в подтверждении объект всех утверждений. Если элемент `<Subject>` пропущен, то утверждения в подтверждении относятся к объекту или объектам, определенным средствами приложения или профиля. Сам язык SAML не определяет таких утверждений, и подтверждение без объекта не имеет в настоящей Рекомендации определенного смысла.

В зависимости от требований конкретных протоколов или профилей, зачастую, может требоваться аутентификация самого создателя подтверждения SAML, также часто может требоваться защита целостности. Аутентификация и целостность сообщения могут быть обеспечены механизмами, протокольных связей, используемыми в ходе доставки подтверждения (см. раздел 10). Подтверждение SAML может быть подписано, что обеспечивает и аутентификацию создателя и защиту целостности.

Если такая подпись используется, то должен быть представлен элемент `<ds:Signature>`, и доверяющая сторона должна подтвердить, что эта подпись корректна (т. е. что подтверждение не подделано) в соответствии с Правилами подписи XML Консорциума W3C. Если она не корректна, то доверяющая сторона не должна доверять содержанию подтверждения. Если она корректна, то доверяющая сторона должна исследовать подпись на предмет ее идентификации и приемлемости создателя и может продолжить обработку подтверждения в соответствии с настоящей Рекомендацией и в том виде, который представляется приемлемым (например, оценить условия, совет, выполнить правила для данного профиля и т. д.).

Вне зависимости от того является ли оно подписанным или неподписанным, введение в одно подтверждение нескольких утверждений семантически эквивалентно множеству отдельных подтверждений, содержащих эти утверждения (при условии, что они относятся к одному и тому же объекту, условиям и т. д.).

Приведенный далее фрагмент схемы определяет элемент `<Assertion>` и его сложный тип **AssertionType**:

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

#### 8.1.3.4 Элемент <EncryptedAssertion>

Элемент <EncryptedAssertion> представляет собой зашифрованное подтверждение, как определено Правилами шифрования W3C. Элемент <EncryptedAssertion> содержит следующие элементы:

– <xenc:EncryptedData> [Требуемый]

  Зашифрованное содержание и соответствующие данные шифрования, определенные Правилами шифрования W3C. Желательно, чтобы был представлен атрибут "Тип" (Type) и, если он представлен, он должен содержать значение http://www.w3.org/2001/04/xmlenc#Element. Зашифрованное содержание должно содержать элемент типа **AssertionType** или типа, являющегося производным от этого типа.

– <xenc:EncryptedKey> [Ноль или несколько]

  Свернутые ключи шифрования, как определено Правилами шифрования W3C. Каждый свернутый ключ должен содержать атрибут Recipient, который определяет элемент, для которого был зашифрован ключ. Значением атрибута Recipient должен быть идентификатор URI элемента системы SAML, как определено в 8.7.

Зашифрованные подтверждения выполняют функции механизма обеспечения конфиденциальности, когда обычные текстовые сообщения передаются через посредника.

Приведенный далее фрагмент схемы определяет элемент <EncryptedAssertion>:

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

### 8.1.4 Объекты

В настоящем разделе определяются конструкции языка SAML, используемые для описания объекта подтверждения. Дополнительный элемент <Subject> определяет клиента, который является объектом всех (нуля или нескольких) утверждений в подтверждении. Он содержит идентификатор, набор из одного или нескольких подтверждений для объекта, либо и то и другое:

– <BaseID>, <NameID> или <EncryptedID> [Дополнительный]

  Идентифицирует объект.

– <SubjectConfirmation> [Ноль или несколько]

  Информация, которая позволяет подтвердить объект. Если представлено несколько подтверждений объекта, тогда достаточно любого из них, если оно соответствует требованиям для подтверждения объекта с целью, для которой применяется данное подтверждение.

Элемент <Subject> может содержать и идентификатор и ноль или несколько подтверждений, которые доверяющая сторона может проверить в процессе обработки подтверждения. Если проверяется любое из вложенных подтверждений объекта, доверяющая сторона может рассматривать элемент, содержащий подтверждение, как элемент, который подтверждающая сторона ассоциирует с клиентом, идентифицированным идентификатором имени и связанным с утверждениями в подтверждении. Этот проверяющий элемент и сам объект могут быть одним и тем же элементом, и могут быть разными элементами.

Если ни одного подтверждения объекта не включено, то связь между тем, кто представляет подтверждения и действительным объектом не определена.

Элемент <Subject> не должен идентифицировать более чем одного клиента.

Приведенный далее фрагмент схемы определяет элемент <Subject> и его сложный тип **SubjectType**:

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

#### 8.1.4.1 Элемент <SubjectConfirmation>

Элемент `<SubjectConfirmation>` предоставляет доверяющей стороне средства для проверки соответствия объекта подтверждения стороне, с которой связывается доверяющая сторона. Он содержит следующие атрибуты и элементы:

– `Method` [Требуемый]

  Ссылка на URI, который идентифицирует протокол или механизм, который должен использоваться для подтверждения объекта. Ссылка на URI, идентифицирующий методы подтверждения, определенные в языке SAML, рассматривается в разделе 11. Дополнительные методы могут быть добавлены путем определения новых URI или по особому соглашению сторон.

– `<BaseID>`, `<NameID>` или `<EncryptedID>` [Дополнительный]

  Идентифицирует элемент, который, как ожидается, будет удовлетворять вложенным требованиям по подтверждению объекта.

– `<SubjectConfirmationData>` [Дополнительный]

  Дополнительная подтверждающая информация, которая должна использоваться в определенном методе подтверждения. Например, обычное содержание этого элемента может быть элементом `<ds:KeyInfo>`, как определено Правилами шифрования W3C, который идентифицирует ключ шифрования (см. также 8.1.4.3). Конкретные методы подтверждения могут определить тип схемы для описания элементов, атрибутов или контента, которые могут появиться в элементе `<SubjectConfirmationData>`.

Приведенный далее фрагмент схемы определяет элемент `<SubjectConfirmation>` и его сложный тип **SubjectConfirmationType**:

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

#### 8.1.4.2 Элемент <SubjectConfirmationData>

Элемент `<SubjectConfirmationData>` имеет сложный тип **SubjectConfirmationDataType**. Он определяет дополнительные данные, которые позволяют подтвердить объект, либо ограничивает условия, при которых может быть выполнено подтверждение объекта. Подтверждение объекта выполняется, когда доверяющая сторона желает проверить связь между элементом, предоставляющим подтверждение (т. е. проверяющим элементом) и объектом, для которого запрошено подтверждение. Он содержит следующие дополнительные атрибуты, которые применимы для любого метода:

– `NotBefore` [Дополнительный]

  Момент времени, до которого объект не может быть подтвержден. Значение времени кодируется в единицах UTC, как описано в 7.3.

– `NotOnOrAfter` [Дополнительный]

  Момент времени, начиная с которого объект не может быть более подтвержден. Значение времени кодируется в единицах UTC, как описано в 7.3.

– `Recipient` [Дополнительный]

  URI, определяющий элемент или местоположение, на которые проверяющий элемент может передать подтверждение. Например, этот атрибут может указывать, что подтверждение должно быть доставлено на определенную оконечную точку сети для того, чтобы не дать возможности посреднику перенаправить его куда либо еще.

– `InResponseTo` [Дополнительный]

  Идентификатор `ID` протокольного сообщения SAML, в ответ на которое проверяющий элемент может направить подтверждение. Например, этот атрибут может использоваться для установления связи между подтверждением и запросом SAML, который привел к созданию подтверждения.

– `Address` [Дополнительный]

  Сетевой адрес/местоположение, откуда проверяющий элемент может направить подтверждение. Например, этот атрибут может использоваться для установления связи между подтверждением и конкретным клиентом для того, чтобы не дать злоумышленнику возможности просто украсть это подтверждение и представить подтверждение, сформированное в другом месте. Адреса IPv4 должны быть представлены в обычном десятичном формате с точкой (например, "1.2.3.4"). Адреса IPv6 должны быть представлены, как определено в документе IETF RFC 3513, 2.2 (например, "FEDC:BA98:7654:3210: FEDC:BA98:7654:3210").

– Необязательные атрибуты

  Этот сложный тип использует точку расширения `<xs:anyAttribute>` для того, чтобы дать возможность добавить в конструкцию `<SubjectConfirmationData>` необязательные атрибуты XML, определенные областью имен, без необходимости иметь явное расширение схемы. Это позволяет добавить дополнительные поля, необходимые для представления дополнительной информации подтверждения. Расширения SAML не должны добавлять местные (не определенные областью имен) атрибуты XML или атрибуты XML, определенные областью имен языка SAML, в сложный тип **SubjectConfirmationDataType** или в типы, являющиеся его производными; такие атрибуты предназначены для будущего поддержания и расширения самого языка SAML.

– Необязательные элементы

  Этот сложный тип использует точку расширения `<xs:any>`, для того чтобы дать возможность добавить в конструкцию `<SubjectConfirmationData>` необязательные элементы XML без необходимости иметь явное расширение схемы. Это позволяет добавить дополнительные элементы , необходимые для представления дополнительной информации подтверждения.

Конкретные методы подтверждения и профили, которые должны использовать эти методы, могут требовать применения одного или нескольких атрибутов, определенных внутри этого сложного типа. Примеры того, как могут использоваться эти атрибуты (и в общем случае подтверждение объекта), рассматривается в разделе 13.

Период времени, определенный дополнительными атрибутами `NotBefore` и `NotOnOrAfter`, если представлен, должен попасть в соответствующий общий период подтверждения достоверности, который определяется атрибутами `NotBefore` и `NotOnOrAfter` элемента `<Conditions>`. Если оба атрибута представлены, то значение `NotBefore` должно быть меньше (раньше) значения `NotOnOrAfter`.

Приведенный далее фрагмент схемы определяет элемент `<SubjectConfirmationData>` и его сложный тип **SubjectConfirmationDataType**:

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

#### 8.1.4.3 Сложный тип KeyInfoConfirmationDataType

Сложный тип **KeyInfoConfirmationDataType** налагает ограничения на элемент `<SubjectConfirmationData>`, разрешая ему содержать один или несколько элементов `<ds:KeyInfo>`, которые определяют ключи шифрования, используемые для аутентификации проверяющей стороны. Конкретные методы подтверждения должен определить точный механизм, при помощи которого могут использоваться данные подтверждения. Могут быть также и дополнительные атрибуты, определенные сложными типом **SubjectConfirmationDataType**.

Этот сложный тип, или тип, являющийся его производным, должен использоваться любым методом подтверждения, который определяет свои данные подтверждения в единицах элемента `<ds:KeyInfo>`.

В соответствии с Правилами шифрования W3C каждый элемент <ds:KeyInfo> должен идентифицировать один-единственный ключ шифрования. Несколько ключей могут быть идентифицированы отдельными элементами <ds:KeyInfo>, как, например, когда клиент использует различные ключи для подтверждения своей личности для различных доверяющих сторон.

Приведенный далее фрагмент схемы определяет сложный тип **KeyInfoConfirmationDataType**:

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

#### 8.1.4.4 Пример элемента <Subject>, подтверждаемого при помощи ключа

Для иллюстрации того, как соответствуют друг другу различные элементы и типы, ниже приведен пример элемента <Subject>, содержащего идентификатор имени и подтверждение объекта, выполняемое на основе владения ключом. Здесь, используя **KeyInfoConfirmationDataType**, подтверждается синтаксис данных подтверждения для элемента <ds:KeyInfo>:

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

### 8.1.5 Условия

В настоящем разделе определяются конструкции языка SAML, которые накладывают ограничения на допустимое использование подтверждения SAML. Элемент <Conditions> может содержать следующие элементы и атрибуты:

– NotBefore [Дополнительный]

  Определяет наиболее ранний момент времени, при котором подтверждение является действующим. Значение времени кодируется в единицах UTC, как описано в 7.3.

– NotOnOrAfter [Дополнительный]

  Определяет момент времени. В который истекает срок действия подтверждения. Значение времени кодируется в единицах UTC, как описано в 7.3.

– <Condition> [Любое число]

  Условие для типа, определенного в схеме расширения. Атрибут xsi:type должен использоваться для указания реального типа условия.

– <AudienceRestriction> [Любое число]

  Указывает, что подтверждение предназначено для определенной аудитории.

– <OneTimeUse> [Дополнительный]

  Указывает, что подтверждение следует использовать сразу же и оно не должно сохраняться для использования в будущем. Несмотря на то что схема позволяет создавать этот элемент несколько раз, одновременно должно существовать не более одного экземпляра этого элемента.

– <ProxyRestriction> [Дополнительный]

  Определяет ограничения, которые подтверждающая сторона накладывает на доверяющие стороны, желающие в последствии сами действовать как подтверждающая сторона и самостоятельно создавать подтверждения на основе информации, содержащейся в исходном подтверждении. Несмотря на то что схема позволяет создавать этот элемент несколько раз, одновременно должно существовать не более одного экземпляра этого элемента.

Поскольку использование атрибута `xsi:type` дало бы возможность иметь внутри подтверждения более одного экземпляра определяемого в языке SAML подтипа типа **ConditionsType** (например, **OneTimeUseType**), схема явным образом не ограничивает число раз, когда могут быть включены определенные условия. Определенный тип условий может определять пределы для такого использования, как показано выше.

Приведенный далее фрагмент схемы определяет элемент `<Conditions>` и его сложный тип **ConditionsType**:

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

#### 8.1.5.1 Общие правила обработки

Если подтверждение содержит элемент `<Conditions>`, то приемлемость подтверждения определяется из имеющихся субэлементов и атрибутов с использованием следующих правил в приведенном ниже порядке.

Подтверждение, которое имеет статус условия корректности "Корректно", может тем не менее не заслуживать доверия или оказаться недействительным по той причине, что оно неправильно сформировано или не соответствует схеме, или создано ненадежным органом SAML, или не аутентифицировано заслуживающими доверия средствами.

Некоторые условия могут не оказывать непосредственного влияния на достоверность передаваемых подтверждений (они всегда оцениваются как Корректные), но могут ограничивать возможности доверяющих сторон относительно применения этого подтверждения:

- если в элементе `<Conditions>` не представлено ни субэлементов, ни атрибутов, то подтверждение считается Корректным для целей обработки условия.
- если какой-либо субэлемент или атрибут в элементе `<Conditions>` определяется как некорректный, то подтверждение считается Некорректным.
- если невозможно оценить какой-либо субэлемент или атрибут в элементе `<Conditions>` или если встречается элемент, который непонятен, то корректность подтверждения определена быть не может, и оно считается неопределенным.
- если все субэлементы или атрибуты в элементе `<Conditions>` определяются как Корректные, то подтверждение считается Корректным для целей обработки условия.

Первое из правил, которое применяется, завершает обработку условий; таким образом, определение, что подтверждение является некорректным, имеет преимущество перед определением того, что оно не определено.

Подтверждение, которое определено, как некорректное или неопределенное, должно быть отброшено доверяющей стороной (внутри любого обрабатываемого контекста или профиля), точно так же, как если бы подтверждение было бы неправильно сформировано или неприменимо.

#### 8.1.5.2 Атрибуты NotBefore и NotOnOrAfter

Атрибуты `NotBefore` и `NotOnOrAfter` определяют пределы времени действия подтверждения внутри используемого контекста или профиля(ей). Они не гарантируют, что в течение периода действия утверждения в подтверждении будут корректными или правильными.

Атрибут `NotBefore` определяет момент времени, когда начинается период действия. Атрибут `NotOnOrAfter` определяет момент времени, когда заканчивается период действия.

Если пропущено значение либо `NotBefore`, либо `NotOnOrAfter`, то оно считается неопределенным. Если не определен атрибут `NotBefore` (и все другие представленные условия определены как Корректные), то подтверждение Корректно для данных условий в любое время до момента, определенного атрибутом `NotOnOrAfter`. Если не определен атрибут `NotOnOrAfter` (и все другие представленные условия определены как Корректные), то подтверждение Корректно для данных условий с момента времени, определенного атрибутом `NotBefore` без истечения срока. Если не определен ни один из этих атрибутов (и если все другие представленные условия определены как Корректные), подтверждение Корректно для данных условий в любое время.

Если представлены оба атрибута, то значение `NotBefore` должно быть меньше, чем (ранее, чем) значения атрибута `NotOnOrAfter`.

#### 8.1.5.3 Элемент <Condition>

Элемент <Condition> используется в качестве точки расширения для новых условий. Его сложный тип **ConditionAbstractType** является абстрактным, и, следовательно, может использоваться только как основа для производного типа.

Приведенный далее фрагмент схемы определяет элемент <Condition> и его сложный тип **ConditionAbstractType**:

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

#### 8.1.5.4 Элементы <AudienceRestriction> и <Audience>

Элемент <AudienceRestriction> указывает, что подтверждение адресовано одной или нескольким конкретным аудиториям, определенным элементами <Audience>. Несмотря на то что доверяющая сторона SAML, которая не входит в число определенной аудитории, имеет возможность делать выводы из данного подтверждения, подтверждающая сторона SAML в явном виде не делает никаких заявлений относительно точности или достоверности этой стороны. Он содержит следующий элемент:

– <Audience>

  Ссылка на URI, которая идентифицирует целевую аудиторию. Ссылка на URI может определять документ, в котором описывается условия участия в аудитории. Он может также содержать уникальный идентификатор URI из идентификатора имени SAML, описывающего элемент системы.

Условия ограничения аудитории считаются Корректными, если и только если доверяющая сторона SAML является членом одной или нескольких определенных аудиторий.

Доверяющая сторона SAML не может помешать стороне, для которой раскрыта информация подтверждения, выполнить действия на основе полученной информации. Однако элемент <AudienceRestriction> позволяет доверяющей стороне SAML явно утверждать, что этой стороне не предоставляется никаких гарантий в виде машинных кодов или в виде читаемого текста. Хотя невозможно гарантировать, что суд поддержал бы такое исключение гарантий в любых условиях, вероятность поддержания исключения гарантий существенно повышена.

В одно-единственное подтверждение может быть включено несколько элементов <AudienceRestriction>, и каждый из них должен быть оценен независимо. Результат этого требования и определения обработки состоит в том, что при данных условиях аудитории образуют дизъюнкцию (объединение с операцией "ИЛИ"), тогда как несколько условий образуют конъюнкцию (объединение с операцией "И").

Приведенный далее фрагмент схемы определяет элемент <AudienceRestriction> и его сложный тип **AudienceRestrictionType**:

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

#### 8.1.5.5 Элемент <OneTimeUse>

Как правило, доверяющие стороны могут выбрать опцию сохранять подтверждения или информацию, которая в них содержится, в некоторой другой форме, для повторного использования. Элемент условий <OneTimeUse> позволяет ответственному органу указать на то, что информация в подтверждении, вероятно, изменится очень скоро и что для каждого использования следуют получать последнюю информацию. В качестве примера можно привести подтверждение, содержащее утверждение <AuthzDecisionStatement>, являющееся результатом политики, которая требовала контроля доступа, зависящего от времени суток.

Если можно точно синхронизировать системные часы в распределенных условиях, то это требование может быть выполнено за счет правильного использования интервала действия. Однако поскольку всегда будет присутствовать некоторая рассинхронизация часов в разных системах, и она будет объединяться с возможными задержками передачи, не существует удобного способа, при помощи которого создатель подтверждения мог бы ограничить срок его действия, не подвергаясь риску, что время жизни подтверждения истечет еще до того, как оно достигнет места назначения.

Элемент <OneTimeUse> указывает, что доверяющая сторона должна использовать подтверждение сразу же, и оно не должно сохраняться для использования в будущем. Доверяющие стороны всегда имеют возможность запросить последнее подтверждение для каждого использования. Однако варианты реализации, которые

принимают решение, сохранят подтверждения для использования в будущем, должны контролировать состояние элемента <OneTimeUse>. Это условие не зависит от информации условий NotBefore и NotOnOrAfter.

Для поддержки ограничения одноразового использования, доверяющая сторона должна хранить кэш-буфер с подтверждениями, которые ею были обработаны, и содержащими такое условие. Когда бы не обрабатывалось подтверждение с таким условием, следует проверить хэш-буфер и убедиться в том, что точно такое подтверждение не было ранее получено и обработано доверяющей стороной.

Ответственный орган SAML не должен вводить в элемент подтверждения <Conditions> более одного элемента <OneTimeUse>.

С целью определения достоверности элемента <Conditions>, элемент <OneTimeUse> всегда считается достоверным. То есть это условие не влияет на достоверность, но является условием использования.

Приведенный далее фрагмент схемы определяет элемент <OneTimeUse> и его сложный тип **OneTimeUseType**:

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

#### 8.1.5.6 Элемент <ProxyRestriction>

Определяет ограничения, которые доверяющая сторона накладывает на доверяющие стороны, которые, в свою очередь, желают в последствии сами действовать как подтверждающая сторона и самостоятельно создавать подтверждения на основе информации, содержащейся в исходном подтверждении. Доверяющая сторона, действующая как подтверждающая сторона, не должна создавать подтверждения, которое нарушало бы ограничения, определенные в этих условиях на основе подтверждения, содержащего такое условие.

Элемент <ProxyRestriction> содержит следующие элементы и атрибуты:

– Count [Дополнительный]

  Определяет максимальное количество непрямых операций, которое доверяющая сторона разрешает иметь между данным подтверждением и подтверждением, которое, в конечном итоге, создается на его основе.

– <Audience> [Ноль или несколько]

  Определяет множество аудиторий, для которых доверяющая сторона разрешает создавать новые подтверждения на основе этого подтверждения.

Значение Count = 0 показывает, что доверяющая сторона не должна создавать подтверждений для другой доверяющей стороны на основе этого подтверждения. Если это значение больше нуля, то любые создаваемые подтверждения должны содержать элемент <ProxyRestriction> со значением Count равное не более чем это значение минус один.

Если не определено ни одного элемента <Audience>, то на доверяющие стороны не налагается ограничений относительно аудиторий, для которых могут быть созданы последующие подтверждения. В противном случае любые подтверждения, созданные таким образом, должны сами содержать элемент <AudienceRestriction>, в составе которого будет как минимум один элемент <Audience>, представленный в предшествующем элементе <ProxyRestriction>, и ни одного элемента <Audience>, которого не было представлено в предшествующем элементе <ProxyRestriction>.

Ответственный орган SAML не должен включать более одного элемента <ProxyRestriction> в элемент подтверждения <Conditions>.

С целью определения достоверности элемента <Conditions>, условие <ProxyRestriction> всегда считается достоверным. То есть это условие не влияет на достоверность, но является условием использования.

Приведенный далее фрагмент схемы определяет элемент <ProxyRestriction> и его сложный тип **ProxyRestrictionType**:

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

### 8.1.6 Элемент Advice

В настоящем разделе определяются конструкции языка SAML, которые содержат дополнительную информацию относительно подтверждения, которую подтверждающая сторона желает предоставить доверяющей стороне.

Элемент <Advice> содержит любую дополнительную информацию, которую желает предоставить ответственный орган SAML. эта информация может быть проигнорирована приложениями без влияния на семантику или достоверность подтверждения.

Элемент <Advice> содержит множество из нуля или нескольких элементов <Assertion>, <EncryptedAssertion>, <AssertionIDRef> и <AssertionURIRef>, и элемент, определенных областью имен в других (не-SAML) областях имен.

Далее приведены некоторые возможные варианты использования элемента <Advice>:

- включать доказательства, поддерживающие утверждения в подтверждении, которые должны цитироваться, либо непосредственно (путем введения утверждений) либо косвенно (при помощи ссылок на поддерживающие подтверждения);
- выражать доказательство запроса подтверждения;
- определять время и точки распределения для обновлений подтверждения.

Приведенный далее фрагмент схемы определяет элемент <Advice> и его сложный тип **AdviceType**:

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

### 8.1.7 Утверждения

Все утверждения, определенные в языке SAML, связаны с объектом. Подтверждения SAML обычно формируются относительно **объекта**, представленного элементом <Subject>. Однако элемент <Subject> является дополнительным, и другие спецификации и профили могут использовать структуру подтверждения SAML для создания аналогичных утверждений без указания объекта, или, возможно, указывая объект иным способом. Приведенные далее подклассы определяют конструкции языка SAML, которые содержат информацию об утверждениях.

#### 8.1.7.1 Элемент <Statement>

Элемент <Statement> представляет собой точку расширения, которая позволяет другим приложениям, основанным на подтверждениях, повторно использовать структуру подтверждения языка SAML. Сам язык SAML получает из точки расширения свои основные утверждения. Его сложный тип **StatementAbstractType** является абстрактным и, следовательно, может использоваться только как основа для производного типа.

Приведенный далее фрагмент схемы определяет элемент <Statement> и его сложный тип **StatementAbstractType**:

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

#### 8.1.7.2 Элемент <AuthnStatement>

Элемент <AuthnStatement> описывает утверждение ответственного органа SAML, подтверждающего, что в определенное время при помощи определенных средств была выполнена аутентификация объекта подтверждения. Подтверждения, содержащие элемент <AuthnStatement>, должны содержать элемент <Subject>.

Он имеет тип **AuthnStatementType**, который расширяет тип **StatementAbstractType** путем добавления следующих элементов и атрибутов:

ПРИМЕЧАНИЕ. – Доля версии V2.0 языка SAML элемент <AuthorityBinding> и его соответствующий тип были удалены из элемента <AuthnStatement>.

– AuthnInstant [Требуемый]

  Определяет время, когда производится аутентификация. Значение времени кодируется в единицах UTC, как описано в 7.3.

– `SessionIndex` [Дополнительный]

  Определяет индекс конкретного сеанса связи между клиентом, идентифицированным данным объектом, и ответственным органом аутентификации.

– `SessionNotOnOrAfter` [Дополнительный]

  Определяет момент времени, когда сеанс связи между клиентом, идентифицированным данным объектом, и ответственным органом SAML, создающим этот элемент, должен считаться законченным. Значение времени кодируется в единицах UTC, как описано в 7.3. Не существует требований по взаимосвязи между этим атрибутом и атрибутом условия `NotOnOrAfter`, который может быть представлен в подтверждении.

– `<SubjectLocality>` [Дополнительный]

  Определяет доменное имя DNS и IP-адрес для системы, от которой была получена аутентификация объекта подтверждения.

– `<AuthnContext>` [Требуемый]

  Контекст, используемый ответственным органом по аутентификации до момента аутентификации включительно, который приводит к формированию этого утверждения. Содержит ссылку на класс контекста аутентификация, объявление правил аутентификации или ссылку на это объявление, или и то и другое. Полное описание информации контекста аутентификация содержится в разделе 12.

Как правило, в качестве значения `SessionIndex` может использоваться любое строчное значение. Однако когда имеет значение секретность, необходимо гарантировать, что значение `SessionIndex` не нарушает работу механизмов обеспечения секретности. Соответственно, это значение не должно использоваться для согласования действий клиента среди участников различных сеансов связи. Далее приведено два варианта решения, которые достигают этой цели и рекомендуются для использования:

- использовать для значений `SessionIndex` малые положительные целые числа (или постоянные значения в списке). Ответственный орган SAML должен определить диапазон значений, значение любого целого числа было бы достаточно большим для предотвращения того, чтобы определенные действия участника были бы согласованы среди участников различных сеансов связи. Ответственный орган SAML должен выбирать значения `SessionIndex` из этого диапазона случайным образом (за исключением тех случаев, когда требуется обеспечить уникальные значения для последовательных элементов, предоставляемых одному и тому же участнику сеанса связи, но как часть отдельного сеанса связи);
- использовать в `SessionIndex` сложенное значение `ID` подтверждения.

Приведенный далее фрагмент схемы определяет элемент `<AuthnStatement>` и его сложный тип **AuthnStatementType**:

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

#### 8.1.7.2.1 Элемент <SubjectLocality>

Элемент `<SubjectLocality>` определяет доменное имя DNS и IP-адрес для системы, от которой была получена аутентификация объекта подтверждения. Он имеет следующие атрибуты:

– `Address` [Дополнительный]

  Сетевой адрес системы, при помощи которой был аутентифицирован клиент, идентифицированный данным объектом. IPv4 адреса должны быть представлены в виде десятичного числа с точкой (например, "1.2.3.4"). IPv6 адреса должны быть представлены как определено в документе IETF RFC 3513, 2.2 (например, "FEDC:BA98:7654:3210:FEDC:BA98:7654:3210").

– `DNSName` [Дополнительный]

  Имя DNS системы, при помощи которой был аутентифицирован клиент, идентифицированный данным объектом.

Этот элемент является полностью информативным, поскольку оба этих поля довольно легко "подделать", но может содержать полезную информацию для ряда приложений.

Приведенный далее фрагмент схемы определяет элемент `<SubjectLocality>` и его сложный тип **SubjectLocalityType**:

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

#### 8.1.7.2.2 Элемент <AuthnContext>

Элемент `<AuthnContext>` определяет контекст события аутентификации. Элемент может содержать ссылку на класс контекста аутентификации, объявление правил аутентификации или объявление, или и то и другое. Его сложный тип **AuthnContextType** имеет следующие элементы:

– `<AuthnContextClassRef>` [Дополнительный]

  Ссылка на URI, идентифицирующий класс контекста аутентификации, который описывает объявление правил аутентификации, которое следует после него.

– **<**`AuthnContextDecl`**>** или `<AuthnContextDeclRef>` [Дополнительный]

  Либо объявление правил аутентификации, представленное этим значением, либо ссылка на URI, который идентифицирует это объявление. Ссылка на URI может непосредственно указывать на документ XML, содержащий ссылку на объявление.

– **<**`AuthenticatingAuthority`**>** [Ноль или несколько]

  Ноль или несколько уникальных идентификаторов ответственных органов аутентификации, которые участвовали в аутентификации клиента (не включая создателя подтверждения, который, предполагается, принимает участие без указания здесь в явном виде).

Полное описание информации контекста аутентификация содержится в разделе 12.

Приведенный далее фрагмент схемы определяет элемент `<AuthnContext>` и его сложный тип **AuthnContextType**:

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

#### 8.1.7.3 Элемент <AttributeStatement>

Элемент <AttributeStatement> описывает утверждение ответственного органа SAML, подтверждающего что объект подтверждения ассоциирован с определенными атрибутами. Подтверждения, содержащие <AttributeStatement>, должны содержать элемент <Subject>.

Он имеет тип **AttributeStatementType**, который расширяет тип **StatementAbstractType** путем добавления следующих элементов:

– <Attribute> или <EncryptedAttribute> [Один или несколько]

  Элемент <Attribute> определяет атрибут объекта подтверждения. Зашифрованный атрибут SAML может быть включен в элемент <EncryptedAttribute>.

Приведенный далее фрагмент схемы определяет элемент <AttributeStatement> и его сложный тип **AttributeStatementType**:

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

##### 8.1.7.3.1 Элемент <Attribute>

Элемент <Attribute> идентифицирует атрибут по его имени и, дополнительно, содержит его значение(я). Он имеет сложный тип **AttributeType**. Он используется внутри утверждения атрибута для выражения конкретных атрибутов и значений, связанных с объектом подтверждения, как описано в предыдущем разделе. Он также используется в запросе на получение атрибутов, определенном в языке SAML. Элемент <Attribute> содержит следующие атрибуты XML:

– Name [Требуемый]

  Имя атрибута.

– NameFormat [Дополнительный]

  Ссылка на URI, представляющая классификацию имени атрибута с целью интерпретации имени. Некоторые ссылки на URI, которые могут использоваться в качестве значений атрибута NameFormat, и соответствующие им описания и правила обработки приведены в 8.7.2. Если значений NameFormat не представлено, то действует идентификатор urn:oasis:names:tc:SAML:2.0:attrname-format:unspecified

– FriendlyName [Дополнительный]

  Строка, которая приводит имя атрибута в форме, понятной для человека, она может быть полезной в тех случаях, когда реальное Name является сложным или скрытым, как, например, OID (определенный в Рекомендации МСЭ-Т X.660) или UUID (определенный в Рекомендации МСЭ-Т X.667). Это значение атрибута не должно использоваться в качестве основы для формальной идентификации атрибутов SAML.

– Произвольно выбранные атрибуты

  Этот сложный тип использует точку расширения <xs:anyAttribute> для того, чтобы получить возможность добавления атрибутов XML в конструкции <Attribute> без необходимости иметь явное расширение схемы. Это позволяет добавить дополнительные поля необходимые для представления дополнительных параметров, которые должны быть использованы, например, в запросе атрибута. Расширения SAML не должны добавлять местные (не определенные областью имен) атрибуты XML или атрибуты XML, определенные областью имен языка SAML, в сложный тип **AttributeType** или в типы, являющиеся его производными; такие атрибуты предназначены для будущего поддержания и расширения самого языка SAML.

– <AttributeValue> [Любое число]

  Содержит значение атрибута. Если атрибут содержит несколько дискретных значений, то рекомендуется, чтобы каждое значение появлялось в своем собственном элементе <AttributeValue>. Если для атрибута предоставляется несколько элементов <AttributeValue>, и какие-либо элементы имеют тип данных, назначенный при помощи xsi:type, то все элементы <AttributeValue> должны иметь идентичный назначенный тип данных.

Значение элемента <Attribute>, который не содержит элементов <AttributeValue>, зависит от его контекста. Если в утверждения <AttributeStatement> атрибут SAML существует, но не имеет значений, то элемент <AttributeValue> должен быть пропущен. Отсутствие значений в запросе <samlp:AttributeQuery> указывает, что запрашивающую сторону интересуют любые или все из названных значений атрибута (см. также 8.2).

Любые другие варианты использования элемента <Attribute> в профилях или других спецификациях должны иметь семантику по определению или пропуску элементов <AttributeValue>.

Приведенный далее фрагмент схемы определяет элемент <Attribute> и его сложный тип **AttributeType**:

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

Элемент <AttributeValue> содержит значение определяемого атрибута SAML. Он имеет тип **xs:anyType**, который позволяет любому корректному XML коду выступить в роли содержания элемента.

Если информационное содержание элемента <AttributeValue> имеет простой тип схемы XML (например, **xs:integer** или **xs:string**), этот тип данных может быть объявлен в явном виде при помощи объявления xsi:type в элементе <AttributeValue>. Если значение атрибута содержит структурированные данные, то необходимые элементы данных могут быть определены в схеме расширения.

ПРИМЕЧАНИЕ. – Указание в <AttributeValue> типов данных, отличных от простого типа схемы XML с использованием xsi:type потребует наличия схемы расширения, которая определяла бы тип данных с целью выполнения обработки схемы.

Если атрибут SAML содержит пустое значение, например, пустую строку, соответствующий элемент <AttributeValue> должен быть пустым (обычно это повторяется в виде <AttributeValue/>). Это позволяет обойти требование, изложенное в 7.1, согласно которому строчные значения в содержании SAML должны содержать как минимум один символ, не являющийся пробелом.

Если атрибут SAML содержит значение "ноль", соответствующий элемент <AttributeValue> должен быть пустым и должен содержать обратный атрибут XML xsi:nil со значением "true" или "1".

Приведенный далее фрагмент схемы определяет элемент <AttributeValue>:

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

#### 8.1.7.3.2 Элемент <EncryptedAttribute>

Элемент <EncryptedAttribute> представляет атрибут SAML в зашифрованном виде, как определено Правилами шифрования W3C. Элемент <EncryptedAttribute> содержит следующие элементы:

– <xenc:EncryptedData> [Требуемый]

  Зашифрованное содержание и соответствующие данные шифрования, как определено Правилами шифрования W3C. Должен быть представлен атрибут "Тип", и, если он представлен, должен содержать значение http://www.w3.org/2001/04/xmlenc#Element. Зашифрованное содержание должно содержать элемент, тип которого является производным от типа **AttributeType**.

– <xenc:EncryptedKey> [Ноль или несколько]

  Свернутые ключи шифрования, как определено Правилами шифрования W3C. Каждый свернутый ключ должен содержать атрибут Recipient, который определяет элемент, для которого был зашифрован ключ. Значением атрибута Recipient должен быть идентификатор URI элемента системы с идентификатором имени SAML, как определено в 8.7.

Зашифрованные атрибуты выполняют функции механизма обеспечения конфиденциальности, когда обычные текстовые сообщения передаются через посредника.

Приведенный далее фрагмент схемы определяет элемент <EncryptedAttribute>:

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

#### 8.1.7.4 Элемент <AuthzDecisionStatement>

Элемент `<AuthzDecisionStatement>` описывает утверждение ответственного органа SAML, подтверждающего, что запрос от объекта подтверждения на доступ к определенному ресурсу привел к принятию указанного решения авторизации на основании некоторых дополнительно определенных доказательств. Подтверждения, содержащие элементы `<AuthzDecisionStatement>`, должны содержать элемент `<Subject>`.

Ресурс, определенный при помощи ссылки на URI. Для того чтобы подтверждение было бы корректно и безопасно интерпретировано, ответственный орган SAML и доверяющая сторона SAML должны одинаковым образом интерпретировать каждую ссылку на URI. Невозможность одинаковой интерпретации ссылки на URI может привести к тому, что в зависимости от кодирования ссылки на URI для данного ресурса будут приниматься различные решения авторизации. Правило по нормализации ссылок на URI содержатся в документе IETF RFC 2396 (раздел 6).

Во избежание неоднозначности из-за различий в кодировании URI, элементы системы SAML должны, по возможности, использовать нормализованную форму URI следующим образом:

- ответственные органы SAML должны кодировать все ссылки на URI ресурсов в нормализованной форме;
- доверяющие стороны до обработки должны преобразовывать ссылки на URI ресурсов в нормализованную форму.

Несогласованная интерпретация ссылок на URI может также быть результатом различий между синтаксисом ссылки на URI и семантикой соответствующей файловой системы. Особого внимания требует случай, когда ссылки на URI используются для определения языка стратегии управления. В системе, использующей подтверждения SAML, должны выполняться приведенные далее условия обеспечения безопасности:

- участки синтаксиса ссылок на URI чувствительны к регистру. Если соответствующая файловая система также чувствительна к регистру, запрашивающая сторона не должна иметь возможность получения ресурса, в котором ей было отказано, изменив регистр в части ссылки на URI данного ресурса;
- многие файловые системы поддерживают такие механизмы, как логические пути и символьные строки, которые дают пользователям возможность установить логическое соответствие между записями в файловой системе. Запрашивающая сторона не должна иметь возможность получения ресурса, в котором ей было отказано, создавая такое соответствие.

Элемент `<AuthzDecisionStatement>` имеет тип **AuthzDecisionStatementType**, который расширяет тип **StatementAbstractType,** добавляя следующие элементы и атрибуты:

– `Resource` [Требуемый]

Ссылка на URI, идентифицирующий ресурс, к которому запрашивается авторизация доступа. Этот атрибут может иметь значение пустой ссылки на URI (""), и его значение определяется как "начало текущего документа", как определено в документе IETF RFC 2396, 4.2.

– `Decision` [Требуемый]

Решение, принятое ответственным органом SAML по отношению к определенному ресурсу. Это значение имеет простой тип **DecisionType**.

– `<Action>` [Один или несколько]

Множество действий, которые разрешено выполнить над определенным ресурсом.

– `<Evidence>` [Дополнительный]

Множество подтверждений того, на которые опирается ответственный орган SAML при принятии этого решения.

Приведенный далее фрагмент схемы определяет элемент `<AuthzDecisionStatement>` и его сложный тип **AuthzDecisionStatementType**:

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

#### 8.1.7.4.1 Простой тип DecisionType

Простой тип **DecisionType** определяет возможные значения, которые должны быть указаны в качестве основания для утверждения о решении по авторизации.

– `Permit`

  Определенное действие разрешено.

– `Deny`

  Определенное действие запрещено.

– `Indeterminate`

  Ответственный орган SAML не может определить, разрешено или запрещено определенное действие.

Значение решения `Indeterminate` используется в тех ситуациях, когда ответственному органу SAML требуется возможность создать утверждающее заявление, но он не способен принять решение. Дополнительная информация, являющаяся причиной отказа или невозможности принять решение, может быть представлена в виде элемента `<StatusDetail>` в соответствующем сообщении `<Response>`.

Приведенный далее фрагмент схемы определяет простой тип **DecisionType**:

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

#### 8.1.7.4.2 Элемент <Action>

Элемент `<Action>` определяет действие над определенным ресурсом, для которого запрашивается разрешение. Его содержание в виде строки данных представляет собой метку действия над определенным ресурсом, разрешение на выполнение которого запрашивается, и он имеет следующий атрибут:

– `Namespace` [Дополнительный]

  Ссылка на URI, представляющий область имен, в котором должно интерпретироваться название определенного действия. Если этот элемент отсутствует, действует область имен `urn:oasis:names:tc:SAML:1.0:action:rwedc-negation`, определенная в 8.7.

  ПРИМЕЧАНИЕ (информативное). – PE 36 (см. OASIS PE:2006) предлагает заменить вышеприведенный текст следующим:

  `Namespace` [Требуемый]

  Ссылка на URI, представляющий область имен, в котором должно интерпретироваться название определенного действия.

Приведенный далее фрагмент схемы определяет элемент `<Action>` и его сложный тип **ActionType**:

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

#### 8.1.7.4.3 Элемент <Evidence>

Элемент `<Evidence>` содержит одно или несколько подтверждений или ссылок подтверждения, на которые опирается ответственный орган SAML при принятии решения об авторизации. Он имеет сложный тип **EvidenceType**. Он содержит смесь из одного или нескольких следующих элементов:

– **`<AssertionIDRef>`** [Любое число]

  Определяет подтверждение посредством ссылки на значение атрибута `ID` подтверждения.

– `<AssertionURIRef>` [Любое число]

  Определяет подтверждение посредством ссылки на URI.

– `<Assertion>` [Любое число]

  Определяет подтверждение его значением.

– `<EncryptedAssertion>` [Любое число]

  Определяет зашифрованное подтверждение его значением.

Предоставление подтверждения в качестве доказательства может повлиять на соглашение о доверии между доверяющей стороной SAML и ответственным органом SAML, принимающим решение об авторизации. Например, в том случае, когда доверяющая сторона SAML представляет подтверждение в запросе в ответственный орган SAML, ответственный орган SAML может использовать это подтверждение как доказательство, принимая свое решение об авторизации, не помечая подтверждение элемента `<Evidence>` как достоверный для доверяющей стороны или любой другой третьей стороны.

Приведенный далее фрагмент схемы определяет элемент `<Evidence>` и его сложный тип **EvidenceType**:

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

## 8.2 Протоколы SAML

Протокольные сообщения SAML могут создаваться и передаваться с использованием различных протоколов. Связи языка SAML в разделе 10 описывают конкретные средства транспортировки протокольных сообщений с применением существующих и широко используемых транспортных протоколов. Профиль SAML в разделе 11 описывает множество приложений этого протокола, определенных в этом разделе вместе с дополнительными правилами обработки, ограничениями и требованиями, которые упрощают взаимодействие.

Конкретные сообщения SAML запроса и ответа выводятся из общего типа. Запрашивающая сторона передает отвечающей стороне SAML элемент, произведенный от типа **RequestAbstractType**, и отвечающая сторона создает элемент тесно связанный с типом **StatusResponseType** или произведенный от него, как показано на рисунке 8-1.

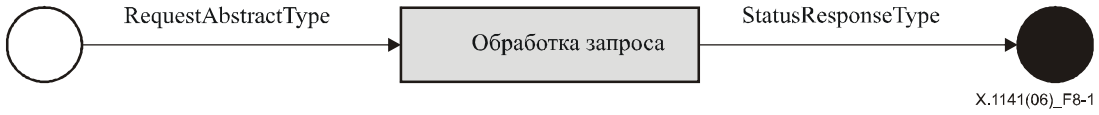


**Рисунок 8-1/X.1141 – Протокол запроса-ответа SAML**

В определенных случаях, когда это разрешено профилями, ответ SAML может быть создан и передан без получения отвечающей стороной соответствующего запроса.

Протоколы, определенные языком SAML, предназначены для выполнения следующих действий:

- возврата одного или нескольких запрошенных подтверждений. Это может выполняться в ответ либо на прямой запрос конкретных подтверждений, либо в ответ на запрос подтверждений, соответствующих определенным критериям;
- выполнения аутентификации по запросу и передачи соответствующего подтверждения;
- регистрации идентификатора имени или отмены регистрации имени по запросу;
- отзыва протокольного сообщения, которое было запрошено по ошибке;
- выполнения почти одновременного отключения от нескольких связанных друг с другом сеансов связи ("единый выход из системы") по запросу;
- выполнения преобразования идентификатора имени по запросу.

В данном разделе текстовые описания элементов и типов в области имен протокола SAML не показаны с обычным префиксом области имен `samlp:`. Для простоты текстовые описания элементов и типов в области имен SAML подтверждения указываются при помощи обычно префикса области имен `saml:`.

### 8.2.1 Объявления заголовка схемы и области имен

Приведенный далее фрагмент схемы определяет области имен XML и другую информацию заголовка для схемы протокола:

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

### 8.2.2 Запросы и ответы

В последующих подразделах определяются конструкции языка SAML и базовые требования, которые лежат в основе всех сообщений запросов и ответов, используемых в протоколах SAML.

#### 8.2.2.1 Сложный тип RequestAbstractType

Все запросы SAML имеют типы, которые являются производными от абстрактного сложного типа **RequestAbstractType**. Этот тип определяет общие атрибуты и элементы, которые связаны со всеми запросами SAML:

ПРИМЕЧАНИЕ. – Для версии V2.0 языка SAML элемент `<RespondWith>` был удален из типа **`RequestAbstractType`**.

– `ID` [Требуемый]

  Идентификатор запроса. Он имеет тип **xs:ID** и должен отвечать требованиям, определенным в 7.4 по уникальности идентификаторов. Значение атрибута `ID` в запросе и атрибута `InResponseTo` в соответствующем ответе должны соответствовать друг другу.

– `Version` [Требуемый]

  Версия данного запроса. Идентификатор для версии SAML, определенной в настоящей Рекомендации имеет значение "2.0".

– `IssueInstant` [Требуемый]

  Момент времени создания запроса. Значение времени кодируется в единицах UTC, как описано в 7.3.

– `Destination` [Дополнительный]

  Ссылка на URI, указывающая адрес, по которому был послан данный запрос. Эта ссылка полезна для предотвращения ошибочного направления запросов адресатам, которым они не предназначены, и для обеспечения защиты, которая требуется для некоторых связей протокола. Если она присутствует, ее получатель должен удостовериться в том, что эта ссылка на URI идентифицирует местоположение, где было получено сообщение. Если это не так, то запрос должен быть отброшен. Некоторые связи протокола могут требовать использования этого атрибута (см. раздел 10).

– `Consent` [Дополнительный]

  Указывает, достигнуто ли согласие (при данных условиях) клиента на передачу этого запроса. Некоторые ссылки на URI, которые могут использоваться в качестве значения атрибута `Consent` и связанные с ними описания, приведены в 8.7.4. Если значения атрибута `Consent` не представлено, действует идентификатор `urn:oasis:names:tc:SAML:2.0:consent:unspecified`.

– `<saml:Issuer>` [Дополнительный]

  Идентифицирует элемент, который создает сообщение-запрос.

– `<ds:Signature>` [Дополнительный]

  Подпись XML, которая аутентифицирует запрашивающую сторону и обеспечивает целостность сообщения, как описано далее и в 8.4.

– `<Extensions>` [Дополнительный]

  Эта точка расширения содержит дополнительные элементы расширения протокола сообщения, которые согласованы между сторонами, поддерживающими связь. Для того чтобы использовать эту точку расширения не требуется никакой схемы расширения, и, даже, если такая схема представлена, неточная установка достоверности не налагает требований по обязательной достоверности этого расширения. Элементы расширения языка SAML должны быть сформированы областью имен в области имен, не определенной в языке SAML.

В зависимости от требований конкретных протоколов или профилей, запрашивающей стороне SAML часто требуется аутентифицировать себя, также часто требуется обеспечить целостность сообщения. Аутентификация и целостность сообщения могут быть обеспечены механизмами, имеющимися в связи протокола (см. раздел 10). Запрос SAML может быть подписан, что обеспечивает и аутентификацию запрашивающей стороны, и целостность сообщения.

Если используется такая подпись, то должен быть представлен элемент `<ds:Signature>` и отвечающая сторона SAML должна убедиться в том, что эта подпись достоверна (т. е. что сообщение не было подделано) в соответствии с Правилами подписи W3C. Если она недостоверна, то отвечающая сторона не должна опираться на содержание этого запроса и должна ответить на него сообщением об ошибке. Если она достоверна, то отвечающая сторона должна оценить подпись для того, чтобы определить ее идентичность и компетенцию подписавшего, и может продолжить обработку запроса или ответить сообщением об ошибке (если запрос недостоверен по каким-либо иным причинам).

Если атрибут `Consent` представлен и его значение указывает, что принципиальное согласие в некоторой форме было достигнуто, то запрос должен быть подписан.

Если отвечающей стороне SAML кажется, что запрос недостоверен в соответствии с синтаксисом SAML или правилами обработки, то, если она отвечает, она должна вернуть сообщение-ответ SAML с элементом `<StatusCode>`, имеющим значение `urn:oasis:names:tc:SAML:2.0:status:Requester`. В некоторых случаях, например во время подозрения на атаку "отказ в обслуживании", может быть рекомендовано не отвечать вообще.

Приведенный далее фрагмент схемы определяет сложный тип **RequestAbstractType**:

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

#### 8.2.2.2 Сложный тип StatusResponseType

Все ответы SAML имеют типы, которые являются производными от абстрактного сложного типа **StatusResponseType**. Этот тип определяет общие атрибуты и элементы, которые связаны со всеми ответами SAML:

– `ID` [Требуемый]

  Идентификатор ответа. Он имеет тип **xs:ID**, и должен отвечать требованиям, определенным в 7.4 по уникальности идентификаторов.

– `InResponseTo` [Дополнительный]

  Ссылка на идентификатор запроса, которому соответствует данный ответ, если таковой имеется. Если ответ создан не в ответ на запрос, или не может быть определено если значение `ID` атрибута запроса (например, запрос неправильно сформирован), то этот атрибут не должен быть представлен. В противном случае он должен быть представлен и его значение должно соответствовать значению соответствующего `ID` атрибута запроса.

– `Version` [Требуемый]

  Версия этого ответа. Идентификатор для Версии языка SAML, определенного в настоящей Рекомендации = "2.0".

– `IssueInstant` [Требуемый]

  Момент времени создания ответа. Значение времени кодируется в единицах UTC, как описано в 7.3.

– `Destination` [Дополнительный]

  Ссылка на URI, указывающая адрес, по которому был послан данный ответ. Эта ссылка полезна для предотвращения ошибочного направления ответов адресатам, которым они не предназначены, и для обеспечения защиты, которая требуется для некоторых связей протокола. Если она присутствует, ее получатель должен удостовериться в том, что эта ссылка на URI идентифицирует местоположение, где было получено сообщение. Если это не так, то ответ должен быть отброшен. Некоторые связи протокола могут требовать использования этого атрибута (см. раздел 10).

– `Consent` [Дополнительный]

  Указывает, достигнуто ли согласие (при данных условиях) клиента на передачу этого ответа. Некоторые ссылки на URI, которые могут использоваться в качестве значения атрибута `Consent` и связанные с ними описания, приведены в 8.7.4. Если значения атрибута `Consent` не представлено, действует идентификатор `urn:oasis:names:tc:SAML:2.0:consent:unspecified` (см. 8.7.4).

– `<saml:Issuer>` [Дополнительный]

  Идентифицирует элемент, который создает сообщение-ответ. (Более подробная информация относительно этого элемента приведена в 8.1.2.5).

– `<ds:Signature>` [Дополнительный]

  Подпись XML, которая аутентифицирует отвечающую сторону и обеспечивает целостность сообщения, как описано далее и в 8.4.

– `<Extensions>` [Дополнительный]

  Эта точка расширения содержит дополнительные элементы расширения протокола сообщения, которые согласованы между сторонами, поддерживающими связь. Для того чтобы использовать эту точку расширения не требуется никакой схемы расширения, и, даже, если такая схема представлена, неточная установка достоверности не налагает требований по обязательной достоверности этого расширения. Элементы расширения языка SAML должны быть сформированы областью имен в области имен, не определенной в языке SAML.

– `<Status>` [Требуемый]

  Код, представляющий собой статус соответствующего запроса.

В зависимости от требований конкретных протоколов или профилей, отвечающей стороне SAML часто требуется аутентифицировать себя, также часто требуется обеспечить целостность сообщения. Аутентификация и целостность сообщения могут быть обеспечены механизмами, имеющимися в связи протокола. Ответ SAML может быть подписан, что обеспечивает и аутентификацию отвечающей стороны, и целостность сообщения.

Если используется такая подпись, то должен быть представлен элемент `<ds:Signature>`, и запрашивающая сторона SAML, получившая ответ, должна убедиться в том, что эта подпись достоверна (т. е. что сообщение не было подделано) в соответствии с Правилами подписи W3C. Если она недостоверна, то запрашивающая сторона не должна опираться на содержание этого ответа и должна рассматривать его как ошибку. Если она достоверна, то запрашивающая сторона должна оценить подпись для того, чтобы определить ее идентичность и компетенцию подписавшего, и может продолжить обработку ответа адекватным образом.

Если атрибут Consent представлен и его значение указывает, что принципиальное согласие в некоторой форме было достигнуто, то ответ должен быть подписан.

Приведенный далее фрагмент схемы определяет сложный тип **StatusResponseType**:

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

**1) Элемент <Status>**

Элемент <Status> содержит следующие элементы:

- <StatusCode> [Требуемый]

  Код, представляющий собой статус действия, выполненного в ответ на соответствующий запрос.

- <StatusMessage> [Дополнительный]

  Сообщение, которое должно быть возвращено оператору.

- <StatusDetail> [Дополнительный]

  Дополнительная информация относительно статуса запроса.

Приведенный далее фрагмент схемы определяет элемент <Status> и его сложный тип **StatusType**:

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

**2) Элемент <StatusCode>**

Элемент <StatusCode> определяет код или множество вложенных кодов, описывающих статус соответствующего запроса. Элемент <StatusCode> содержит следующий элемент и атрибут:

- Value [Требуемый]

  Значение кода статуса. Этот атрибут содержит ссылку на URI. Значение самого старшего элемента <StatusCode> должно быть выбрано их перечня высшего уровня, приведенного в настоящем разделе.

- <StatusCode> [Дополнительный]

  Код подчиненного статуса, которые предоставляет более подробную информацию об условии возникновения ошибки. Отвечающие стороны могут не указывать коды подчиненного статуса для предотвращения атак, которые направлены на получения дополнительной информации путем намеренной передачи ошибочных запросов.

Допустимыми значениями <StatusCode> высшего уровня являются следующие:

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

Запрос успешно обработан. Дополнительная информация быть возвращена в элементах <StatusMessage> и/или <StatusDetail>.

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

Запрос не может быть выполнен из-за ошибки запрашивающей стороны.

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

Запрос не может быть выполнен из-за ошибки отвечающей стороны SAML или ответственного органа SAML.

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

Отвечающая сторона SAML не может обработать запрос из-за того, что Версия сообщения-запроса некорректна.

Приведенные далее коды статуса второго уровня указываются в различных местах настоящей Рекомендации. Дополнительные коды статуса второго уровня могут быть определены в будущих версиях Рекомендации по языку SAML. Элементы системы могут определять более конкретные коды статуса, указывая соответствующие ссылки на URI.

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

Отвечающий провайдер не имел возможности успешно аутентифицировать клиента.

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

В элементе `<saml:Attribute>` или `<saml:AttributeValue>` найдено неожиданное или некорректное содержание.

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

Отвечающий провайдер не может или не будет поддерживать правила запрошенного идентификатора имени.

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

Указанные требования к контексту аутентификации не могут быть выполнены отвечающей стороной.

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

Используется промежуточным элементом для указания того, что ни один из элементов `<Loc>` в списке поддерживаемых провайдеров идентификации `<IDPList>` не может быть распознан или что недоступен ни один из поддерживаемых провайдеров идентификации.

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

Указывает, что отвечающий провайдер не может аутентифицировать клиента при помощи пассивных средств, как было запрошено.

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

Используется промежуточным элементом для указания того, что этим промежуточным элементом не поддерживается ни один из провайдеров идентификации в списке `<IDPList>`.

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

Используется ответственным органом сеанса связи для указания участнику сеанса связи, что невозможно распространить действие окончания сеанса связи на всех остальных участников сеанса связи.

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

Указывает, что отвечающий провайдер не может аутентифицировать клиента непосредственно и не разрешает передавать запрос далее.

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

Отвечающая сторона SAML или ответственный орган SAML могут обработать запрос, но приняли решение не отвечать. Этот код статуса может использоваться, когда имеется беспокойство относительно безопасности контекста сообщения-запроса или последовательности сообщений-запросов, принятых от определенной запрашивающей стороны.

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

Отвечающая сторона SAML или ответственный орган SAML не поддерживает запрос.

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

Отвечающая сторона SAML не может обрабатывать никакие запросы с использованием той версии протокола, которая указаны в запросе.

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

Отвечающая сторона SAML не может обработать запрос, потому что версия протокола, определенная в сообщении-запросе, новее наиболее последней версии протокола, поддерживаемой отвечающей стороной.

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

Отвечающая сторона SAML не может обработать запрос, потому что версия протокола, определенная в сообщении-запросе, слишком старая.

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

Значение ресурса, представленное в сообщении-запросе, недостоверно, или не может быть распознано.

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

Сообщение-ответ должно содержать больше элементов, чем может вернуть отвечающая сторона SAML.

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

Элемент, не имеющий сведений о конкретном профили атрибута, был представлен с атрибутом, полученным из этого профиля.

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

Отвечающий провайдер не может распознать клиента, указанного или подразумеваемого в запросе.

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

Отвечающая сторона SAML не может правильно выполнить запрос, используя связь протокола, указанную в запросе.

Приведенный далее фрагмент схемы определяет элемент <StatusCode> и его сложный тип **StatusCodeType**:

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

**3) Элемент <StatusMessage>**

Элемент <StatusMessage> определяет сообщение, которое может быть возвращено оператору:

Приведенный далее фрагмент схемы определяет элемент <StatusMessage>:

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

**4) Элемент <StatusDetail>**

Элемент <StatusDetail> может использоваться для определения дополнительной информации относительно статуса запроса. Дополнительная информация состоит из нуля или нескольких элементов из любой области имен, без каких-либо требований к содержанию <StatusDetail> по наличию схемы или по проверке достоверности схемы.

Приведенный далее фрагмент схемы определяет элемент <StatusDetail> и его сложный тип **StatusDetailType**:

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

### 8.2.3 Протокол поиска и запроса подтверждения

В настоящем разделе определяются сообщения и правила обработки по запросу существующих подтверждения, путем ссылок или поиска подтверждений при помощи объекта и типа утверждения.

#### 8.2.3.1 Элемент <AssertionIDRequest>

Если запрашивающая сторона знает уникальный идентификатор одного или нескольких подтверждений, то для их запроса может использоваться элемент сообщения <AssertionIDRequest>, в ответ на который будет передано сообщение <Response>. Элемент <saml:AssertionIDRef> используется для определения каждого подтверждения, которое должно быть возвращено.

Приведенный далее фрагмент схемы определяет элемент <AssertionIDRequest>:

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

#### 8.2.3.2 Вопросы

Приведенные далее подклассы определяют сообщения SAML вопросы.

##### 8.2.3.2.1 Элемент <SubjectQuery>

Элемент сообщения `<SubjectQuery>` представляет собой точку расширения, которая позволяет определить новые SAML вопросы, которые определяют один объект SAML. Его сложный тип **SubjectQueryAbstractType** является абстрактным и, следовательно, может использоваться только как основа для производного типа. Тип **SubjectQueryAbstractType** добавляет к типу **RequestAbstractType** элемент `<saml:Subject>` (определенный в 8.1.4).

Приведенный далее фрагмент схемы определяет элемент `<SubjectQuery>` и его сложный тип **SubjectQueryAbstractType**:

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

##### 8.2.3.2.2 Элемент <AuthnQuery>

Элемент сообщения `<AuthnQuery>` используется для передачи вопроса: "Какие подтверждения, содержащие утверждения аутентификации, имеются для данного объекта?" Успешное сообщение `<Response>` будет содержать одно или несколько подтверждений, содержащих утверждения аутентификации.

Сообщение `<AuthnQuery>` не должно использоваться как запрос новой аутентификации, использующие данные, подтверждающие личность, представленные в запросе. `<AuthnQuery>` – это запрос на получение сведений о действиях по аутентификации, которые были выполнены в ходе предшествующего взаимодействия между указанным объектом и ответственным органом по аутентификации.

Этот элемент имеет тип **AuthnQueryType**, который расширяет тип **SubjectQueryAbstractType**, добавляя следующие элемент и атрибут:

– `SessionIndex` [Дополнительный]

  Если он представлен, то определяет фильтр для возможных ответов. Такой вопрос спрашивает: "Какие у Вас есть подтверждения, содержащие утверждения аутентификации, для данного объекта в рамках представленной информации о сеансе связи?"

– `<RequestedAuthnContext>` [Дополнительный]

  Если представлен, определяет фильтр для возможных ответов. Такой вопрос спрашивает "Какие у вас есть подтверждения, содержащие утверждения аутентификации для данного объекта, которые требованиям к контексту аутентификации в этом элементе?"

В ответ на запрос аутентификации, ответственный орган SAML возвращает следующие подтверждения с утверждениями аутентификации:

- правила, приведенные в 8.2.3.4 для определения соответствия с элементом `<Subject>` вопроса, идентифицируют подтверждения, которые могут быть возвращены;
- если в вопросе представлен атрибут `SessionIndex`, то как минимум один элемент `<AuthnStatement>` из множества возвращенных подтверждений должен содержать атрибут `SessionIndex`, который совпадает с атрибутом `SessionIndex` в вопросе. Дополнительным является то, что в ответе должен быть возвращен полный набор всех таких совпадающих подтверждений;
- если в вопросе представлен элемент `<RequestedAuthnContext>`, то как минимум один элемент `<AuthnStatement>` из множества возвращенных подтверждений должен содержать элемент `<AuthnContext>`, который удовлетворяет требованиям элемента в запросе. Дополнительным является то, что в ответе должен быть возвращен полный набор всех таких совпадающих подтверждений.

Приведенный далее фрагмент схемы определяет элемент <AuthnQuery> и его сложный тип **AuthnQueryType**:

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

**1) Элемент <RequestedAuthnContext>**

Элемент <RequestedAuthnContext> определяет требования к контексту аутентификации утверждений аутентификации, возвращаемых в ответ на запрос или вопрос. Его сложный тип **RequestedAuthnContextType** определяет следующие элементы и атрибуты:

– <saml:AuthnContextClassRef> или <saml:AuthnContextDeclRef> [Один или несколько]

  Определяет один или несколько ссылок на URI, идентифицирующий классы контекстов аутентификации или деклараций. Эти элементы определяются в 8.1.7.2.2. Более подробная информация о классах контекстов аутентификации приведена в разделе 12.

– Comparison [Дополнительный]

  Определяет метод сравнения, используемый для оценки запрашиваемых классы контекстов аутентификации или утверждений, может быть одним из следующих "exact", "minimum", "maximum" или "better". Значение по умолчанию = "exact".

Может использоваться либо множество ссылок на класс, либо множество ссылок на декларации, множество имеющихся ссылок должно быть оценено как упорядоченное множество, первым элементом которого является наиболее предпочтительный класс контекста аутентификации или декларация. Если в соответствии с приведенными далее правилами ни один из указанных классов или деклараций не может быть выполнен, то отвечающая сторона должна вернуть сообщение <Response> с кодом второго уровня <StatusCode> для urn:oasis:names:tc:SAML:2.0:status:NoAuthnContext.

Если Comparison имеет значение "exact" или пропущен, то результирующий контекст аутентификации в утверждении аутентификации должен точно совпадать как минимум с одним из определенных контекстов аутентификации.

Если Comparison имеет значение "minimum", то результирующий контекст аутентификации в утверждении аутентификации должен быть как минимум таким же сильным (как его видит отвечающая сторона), насколько силен один из определенных контекстов аутентификации.

Если Comparison имеет значение "better", то результирующий контекст аутентификации в утверждении аутентификации должен быть сильнее (как его видит отвечающая сторона) чем любой из определенных контекстов аутентификации.

Если Comparison имеет значение "maximum", то результирующий контекст аутентификации в утверждении аутентификации должен быть максимально сильным (его видит отвечающая сторона), но не превосходить силы как минимум одного из определенных контекстов аутентификации.

Приведенный далее фрагмент схемы определяет элемент <RequestedAuthnContext> и его сложный тип **RequestedAuthnContextType**:

```
<element name="RequestedAuthnContext" type="samlp:RequestedAuthnContextType"/>
<complexType name="RequestedAuthnContextType">
<choice>
   <element ref="saml:AuthnContextClassRef" maxOccurs="unbounded"/>
   <element ref="saml:AuthnContextDeclRef" maxOccurs="unbounded"/>
</choice>
<attribute name="Comparison" type="samlp:AuthnContextComparisonType"
use="optional"/>
</complexType>
<simpleType name="AuthnContextComparisonType">
<restriction base="string">
   <enumeration value="exact"/>
   <enumeration value="minimum"/>
   <enumeration value="maximum"/>
   <enumeration value="better"/>
</restriction>
</simpleType>
```

#### 8.2.3.2.3 Элемент <AttributeQuery>

Элемент `<AttributeQuery>` используется для запроса "Вернуть запрошенные атрибуты для данного объекта". Успешно выполненный ответ будет иметь форму подтверждений, содержащих утверждения об атрибутах до степени, разрешенной установленными правилами. Этот элемент имеет тип **AttributeQueryType**, который расширяет **SubjectQueryAbstractType**, добавляя следующий элемент:

– `<saml:Attribute>` [Любое число]

Каждый элемент `<saml:Attribute>` определяет атрибут, значение(я) которого должны быть возвращены. Если не определено ни одного атрибута, это значит, что запрашиваются все атрибуты, разрешенные действующими правилами. Если данный элемент `<saml:Attribute>` содержит один или несколько элементов `<saml:AttributeValue>`, то, если в ответе возвращается этот атрибут, он не должен содержат каких-либо значений, которые не равны значениям, определенным в вопросе. При отсутствии правил равенства, установленных конкретными профилями или атрибутами, равенство определяется как идентичное XML представление данного значения. Более подробная информация об атрибуте `<saml:Attribute>` содержится в 8.1.7.3.1.

Один вопрос не должен содержать два элемента `<saml:Attribute>` с одинаковыми значениями `Name` и `NameFormat` (т. е. в вопросе данный атрибут должен быть назван только один раз).

В ответ на запрос атрибута ответственный орган SAML возвращает подтверждения со следующими утверждениями атрибутов:

- Правила, приведенные в 8.2.3.4 для определения соответствия с элементом <Subject> вопроса, идентифицируют подтверждения, которые могут быть возвращены.
- Если в вопросе представлены какие-либо элементы `<Attribute>`, то они ограничивают/фильтруют атрибуты и, дополнительно, значения, как отмечено выше.
- Возвращенные атрибуты и значения могут также быть ограничены соображениями правил, определяемых приложением.

Коды статуса второго уровня `urn:oasis:names:tc:SAML:2.0:status:UnknownAttrProfile` и `urn:oasis:names:tc:SAML:2.0:status:InvalidAttrNameOrValue` могут использоваться для указания проблем с интерпретацией информации атрибута или значения в вопросе.

Приведенный далее фрагмент схемы определяет элемент `<AttributeQuery>` и его сложный тип **AttributeQueryType**:

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

#### 8.2.3.2.4 Элемент <AuthzDecisionQuery>

Элемент `<AuthzDecisionQuery>` используется для передачи вопроса "Должны ли быть разрешены данному объекту эти действия над этими ресурсами, учитывая имеющиеся доказательства?" Успешный ответ будет иметь вид подтверждений, содержащих утверждения решения по авторизации.

ПРИМЕЧАНИЕ. – Функция `<AuthzDecisionQuery>` была заморожена для версии SAML V2.0, и будущих расширений не планируется. Пользователи, которым требуются дополнительные функциональные возможности, могут пожелать рассмотреть возможность использования Расширяемого языка разметки контроля доступа (см. Рек. МСЭ-Т X.1142), который предоставляет возможности расширенного принятия решения об авторизации.

Этот элемент имеет тип **AuthzDecisionQueryType**, который расширяет **SubjectQueryAbstractType** путем добавления следующих элементов и атрибута:

– `Resource` [Требуемый]

Ссылка на URI, идентифицирующий ресурс, к которому запрашивается авторизация доступа.

– `<saml:Action>` [Один или несколько]

Действия, для которых запрашивается авторизация. Более подробная информация по этому элементу содержится в 8.1.7.4.2.

– `<saml:Evidence>` [Дополнительный]

Множество подтверждений, на которые может опираться ответственный орган SAML при принятии решения об авторизации. Более подробная информация об этом элементе содержится в 8.1.7.4.3.

В ответ на запрос решения об авторизации, ответственный орган SAML возвращает подтверждения с утверждениями относительно решений об авторизации следующего вида:

- Правила, приведенные в 8.2.3.4 для определения соответствия с элементом <Subject> вопроса, идентифицируют подтверждения, которые могут быть возвращены.

Приведенный далее фрагмент схемы определяет элемент <AuthzDecisionQuery> и его сложный тип **AuthzDecisionQueryType**:

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

### 8.2.3.3 Элемент <Response>

Элемент сообщения <Response> используется, когда ответ состоит из списка из нуля или нескольких подтверждений, которые удовлетворяют запросу. Он имеет сложный тип **ResponseType**, который расширяет тип **StatusResponseType** и добавляет следующие элементы:

– <saml:Assertion> или <saml:EncryptedAssertion> [Любое число]

Определяет подтверждение его значением или, дополнительно, определяет зашифрованное подтверждение его значением. Более подробная информация об этих элементах содержится в 8.1.3.3.

Приведенный далее фрагмент схемы определяет элемент <Response> и его сложный тип **ResponseType**:

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

### 8.2.3.4 Правила обработки

В ответ на сообщение-запрос, определенный в языке SAML, каждое подтверждение, возвращенное ответственным органом SAML, должно содержать элемент <saml:Subject>, который **точно соответствует** элементу <saml:Subject>, имеющемуся в запросе.

Элемент <saml:Subject> S1 точно соответствует S2, только если выполняются оба следующих условия:

- если S2 содержит элемент идентификатора (<BaseID>, <NameID> или <EncryptedID>), то S1 должен содержать идентичный элемент идентификатора, но этот элемент может быть зашифрован (или нет) либо в виде S1, либо в виде S2. Другими словами, дешифрованные формы идентификаторов S1 и S2 должны быть идентичными. "Идентичный" означает, что идентификатор содержания элемента и значения атрибута должны быть одинаковыми. Согласно этому определению зашифрованный идентификатор после дешифрации будет идентичным исходному;
- если S2 содержит один или несколько элементов <saml:SubjectConfirmation>, то S1 должен содержать как минимум один элемент <saml:SubjectConfirmation>, так чтобы можно было подтвердить S1 описанным способом при помощи как минимум одного элемента <saml:SubjectConfirmation> в S2.

Приведем пример того, что разрешено, и что не разрешено. S1 может содержать <saml:NameID> с конкретным значением Format, а S2 может содержать элемент <saml:EncryptedID>, который является результатом шифрования элемента S1 <saml:NameID>. Однако S1 и S2 не могут содержать элемент <saml:NameID> с разными значениями Format и содержанием элемента даже, если считается, что два этих идентификатора указывают одного и того же клиента.

Если ответственный орган SAML не может предоставить подтверждение с утверждениями, удовлетворяющими ограничениям, выраженным в запросе или в ссылке на подтверждение, элемент <Response> не должен содержать элемент <Assertion> и должен содержать элемент <StatusCode> со значением urn:oasis:names:tc:SAML:2.0:status:Success.

Должны быть учтены все остальные правила обработки, связанные с лежащими в его основе сообщениями запроса и ответа.

### 8.2.4 Протокол запроса аутентификации

Когда клиент (или агент, действующий от имени клиента) желает получить подтверждения, содержащие утверждения аутентификации, для создания режима безопасности на стороне одной или нескольких доверяющих сторон, он может использовать протокол запроса аутентификации для передачи элемента сообщения `<AuthnRequest>` ответственному органу SAML и попросить, чтобы он вернул сообщение `<Response>`, содержащее одно или несколько таких подтверждений. Такие подтверждения могут содержать дополнительные утверждения любого типа, но как минимум одно утверждение должно содержать как минимум одно утверждение аутентификации. Ответственный орган SAML, который поддерживает этот протокол, также называют провайдером идентификации.

Кроме этого требования, конкретное содержание возвращаемых подтверждений зависит от используемого профиля или контекста. Кроме того, точное средство, при помощи которого клиент или агент аутентифицирует себя для провайдер идентификации, не определено, хотя средства аутентификации могут повлиять на содержание ответа. Другие проблемы, связанные с проверкой провайдером идентификации документов, удостоверяющих личность или любой связи между провайдером идентификации и любыми другими элементами, участвующими в процессе аутентификации, также в сферу применения данного протокола не входят.

Описания и правила обработки в последующих подразделах, называют следующие действующие лица, многие из которых могут быть теми же элементами конкретного используемого профиля:

– Запрашивающая сторона

  Элемент, который создает запрос аутентификации и к которому должен вернуться ответ.

– Представитель

  Элемент, который представляет запрос провайдеру идентификации и либо сам аутентифицирует себя во время передачи сообщения, либо надеется на существующий контекст безопасности для установления его идентичности. Если запрашивающей стороны нет, представитель действует как промежуточный элемент между запрашивающей стороной и отвечающим провайдером идентификации.

– Запрошенный `объект`

  Элемент, о котором запрашивается одно или несколько подтверждений.

– Аттестующий элемент

  Элемент или элементы, которые, как ожидается, должны быть способны удовлетворить один из элементов `<SubjectConfirmation>` в результирующем(их) подтверждении(ях).

– Доверяющая сторона

  Элемент или элементы, которые, как ожидается, будут использовать подтверждение(я) для достижения цели, определенной профилем или контекстом использования обычно для установления режима безопасности.

– Провайдер идентификации

  Элемент, которому представитель передает запрос, и от которого представитель получает ответ.

**Элемент <AuthnRequest>**

Для того чтобы попросить провайдера идентификации создать подтверждение с утверждением аутентификации, представитель аутентифицирует себя для этого провайдера идентификации (либо надеется на существующий контекст безопасности) и передает сообщение `<AuthnRequest>`, описывающее свойства, которые должно иметь результирующее подтверждение для того, удовлетворять своей цели. Среди этих свойств может быть информация, которая опирается на содержание подтверждения и/или информация, которая основана на том, как должно быть доставлено запрашивающей стороне результирующее сообщение `<Response>`. Процесс аутентификации представителя может быть выполнен до, во время или после первоначальной доставки сообщения `<AuthnRequest>`.

Запрашивающая сторона может отличаться от стороны, передающей запрос, если, например, запрашивающая сторона является доверяющей стороной, которая стремится использовать результирующее подтверждение для аутентификации или авторизации запрошенного объекта, так что доверяющая сторона может сама выбирать, предоставлять или нет услуги.

Сообщение `<AuthnRequest>` должно быть подписано или в противном случае аутентифицировано, и его целостность должна быть защищена связью протокола, используемого для доставки сообщения.

Это сообщение имеет сложный тип **AuthnRequestType**, который расширяет **RequestAbstractType** и добавляет следующие элементы и атрибуты, все они, как правило, являются дополнительными. Но могут быть потребованы для конкретных профилей:

– `<saml:Subject>` [Дополнительный]

  Определяет запрошенный объект результирующего(их) подтверждения(й). Он может содержать один или несколько элементов `<saml:SubjectConfirmation>` для того, чтобы указать, как и/или при помощи чего могут быть подтверждены результирующие подтверждения. Более подробная информация об этом элементе содержится в 8.1.4.

Если этот элемент полностью пропущен, или если в него не включено ни одного идентификатора, то предполагается, что запрошенным объектом является представитель сообщения. Если не включено ни одного элемента <saml:SubjectConfirmation>, предполагается, что представитель является единственным требуемым аттестующим элементом, и метод ее выполнения определяется используемым профилем и/или правилами провайдера идентификации.

– <NameIDPolicy> [Дополнительный]

Определяет ограничения, накладываемые на идентификатор имени, который должен использоваться для представления запрошенного объекта. Если он пропущен, то может использоваться любой тип идентификатора, поддерживаемый провайдером идентификации для запрошенного объекта, ограниченный любыми соответствующими ситуации правилами, определенными вариантом использования, например, учитывающими требования секретности,.

– <saml:Conditions> [Дополнительный]

Определяет условия SAML, которые ожидает запрашивающая сторона для ограничения срока действия и/или использования полученных подтверждения(й). Отвечающая сторона может изменять или дополнять это множество, если считает это необходимым. Информация в этом элементе используется как исходные данные до процесса создания подтверждения, а не в качестве условий использования самого запроса. (Более подробная информация об этом элементе содержится в 8.1.5.)

– <RequestedAuthnContext> [Дополнительный]

Определяет требования, если таковые имеются, которые запрашивающая сторона налагает на контекст аутентификации, которые применяются к аутентификации представителя сообщения отвечающим провайдером.

– <Scoping> [Дополнительный]

Определяет объем данных для идентификации провайдера, которому запрашивающая сторона доверяет аутентифицировать представителя сообщения, а также ограничения и контекст, относящиеся к посреднической передаче отвечающей стороной сообщения <AuthnRequest> следующим провайдерам идентификации.

– ForceAuthn [Дополнительный]

Булева величина. Если ее значение = "true", то провайдер идентификации должен аутентифицировать представителя непосредственно, а не использовать предыдущий контекст безопасности. Если эта величина не представлена, значение по умолчанию = "false". Однако если обе величины ForceAuthn и IsPassive имеют значение "true", то провайдер идентификации не должен заново аутентифицировать представителя, если выполняются ограничения, определенные в IsPassive.

– IsPassive [Дополнительный]

Булева величина. Если ее значение = "true", то провайдер идентификации и сам агент пользователя не должны иметь видимого контроля над интерфейсом пользователя от запрашивающей стороны и заметно взаимодействовать с тем, кто ее представляет. Если эта величина не представлена, значение по умолчанию = "false".

– AssertionConsumerServiceIndex [Дополнительный]

Косвенно идентифицирует место, куда сообщение <Response> должно быть возвращено запрашивающей стороне. Применяется только для профилей, в которых запрашивающая сторона отличается от представителя, например, профиль SSO веб-браузера, описанный в настоящей Рекомендации. Провайдер идентификации должен обладать доверенными средствами для сопоставления значений индекса в атрибуте местоположению запрашивающей стороны. В разделе 9 описан один из возможных механизмов. Если он пропущен, то провайдер идентификации должен возвратить сообщение <Response> в местоположение запрашивающей стороны, определенное "по умолчанию" для используемого профиля. Если указанный индекс недействителен, то провайдер идентификации может вернуть сообщение об ошибке <Response>, либо он может использовать местоположение, определенное "по умолчанию". Этот атрибут является взаимоисключающим с атрибутами подтверждения ConsumerServiceURL и ProtocolBinding.

– AssertionConsumerServiceURL [Дополнительный]

Определяет своим значением место, сообщение <Response> должно быть возвращено запрашивающей стороне. Отвечающая сторона должна какими-либо средствами гарантировать, что действительно определено значение, ассоциированное с запрашивающей стороной. В разделе 9 описан один из возможных механизмов; другой механизм – подпись завершающего сообщения <AuthnRequest>. Этот атрибут является взаимоисключающим с атрибутом подтверждения ConsumerServiceIndex и, как правило, его сопровождает атрибут ProtocolBinding.

– ProtocolBinding [Дополнительный]

Ссылка на URI, которая идентифицирует связь протокола SAML, которая должна использоваться при возвращении сообщения <Response>. Более подробная информация о связях протокола и определенных для них ссылках на URI содержится в разделе 10. Этот атрибут является взаимоисключающим с атрибутом подтверждения ConsumerServiceIndex и, как правило, его сопровождает атрибут ConsumerServiceURL.

– `AttributeConsumingServiceIndex` [Дополнительный]

Косвенно идентифицирует информацию, связанную с запрашивающей стороной и описывающую атрибуты SAML, которые запрашивающая сторона желательно или необходимо получить от провайдера идентификации в сообщении `<Response>`. Провайдер идентификации должен обладать доверенными средствами для сопоставления значений индекса в атрибуте с информацией, связанной с запрашивающей стороной. В разделе 9 описан один из возможных механизмов. Провайдер идентификации может использовать эту информацию для повторения одного или нескольких элементов `<saml:AttributeStatement>` в подтверждении(ях), которые он возвращает.

– `ProviderName` [Дополнительный]

Определяет понятное человеку название запрашивающей стороны для использования агентом пользователя представителя или провайдером идентификации.

Общие правила обработки для этого сообщение приведены в 8.2.4.4.

Приведенный далее фрагмент схемы определяет элемент `<AuthnRequest>` и его сложный тип **AuthnRequestType**:

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

#### 8.2.4.1 Элемент <NameIDPolicy>

Элемент `<NameIDPolicy>` адаптирует идентификатор имени в объектах подтверждений, полученных из `<AuthnRequest>`. Его сложный тип **NameIDPolicyType** определяет следующий атрибуты:

– `Format` [Дополнительный]

Определяет ссылку на URI, соответствующий формату идентификатора имени, определенному в настоящей или другой Рекомендации (например, см. 8.7.3). Дополнительное значение элемента `urn:oasis:names:tc:SAML:2.0:nameid-format:encrypted` определено специально для применения внутри этого атрибута для указания запроса на шифрование результирующего идентификатора.

– `SPNameQualifier` [Дополнительный]

Дополнительно определяет, что идентификатор объекта подтверждения должен быть возвращен (или создан) в области имен провайдера услуг, отличающегося от запрашивающей стороны, или в области имен объединенной группы провайдеров услуг. Например, в настоящей Рекомендации приведено определение `urn:oasis:names:tc:SAML:2.0:nameid-format:persistent`.

– `AllowCreate` [Дополнительный]

Булева величина, используется для указания того, разрешено ли провайдеру идентификации в ходе выполнения запроса создавать новый идентификатор для описания клиента. Значение по умолчанию = "false". Когда значение этой величины = "false", запрашивающая сторона ограничивает провайдера идентификации, разрешая создавать только подтверждения для себя, если уже имеется идентификатор

для клиента. Это не мешает провайдеру идентификации создавать такие идентификаторы вне рамок данного конкретного запроса (например, заранее для большого количества участников).

ПРИМЕЧАНИЕ 1 (информативное). – PE14 (см. OASIS PE:2006) разъясняет вышеприведенное определение следующим образом:

Булева величина, используемая для указания того, дает ли запрашивающая сторона право провайдеру идентификации, в ходе выполнения запроса, создавать новый идентификатор или ассоциировать существующий идентификатор для описания клиента с доверяющей стороной. Значение по умолчанию = "false", если величина не представлена или пропущен весь элемент.

ПРИМЕЧАНИЕ 2 (информативное). – PE14 (см. OASIS PE:2006) предлагает добавить в нижеприведенный параграф следующий текст:

Атрибут `AllowCreate` может использоваться в ряде вариантов использования для влияния на создание состояния, поддерживаемого провайдером идентификации, относящегося к использованию определенной доверяющей стороной идентификатора имени (или любых других постоянно существующих, уникально идентифицирующих атрибутов) с целью создания динамического идентификатора или атрибута, отслеживания согласованного последовательного использования протокола управления идентификатором имени или для других соответствующих целей.

Когда его значение = "false", запрашивающая сторона стремится ограничить возможности провайдера идентификации создавать подтверждение только, если такое состояние уже было установлено ранее или использование идентификатор не кажется пригодным для провайдера идентификации. Таким образом, это не мешает провайдеру идентификации предположить, что такая информация существует вне контекста данного конкретного запроса (например, создается заранее для большого количества участников).

Значение = "true" позволяет провайдеру идентификации выполнять любые связанные с этим действия для выполнения запроса, учитывая все другие ограничения, накладываемые запросом и атрибутом правил (например, атрибутом `IsPassive`).

Обычно запрашивающие стороны не могут предположить определенных действий со стороны провайдеров идентификации в отношении первоначального создания или связывания идентификаторов от их имени, поскольку эти детали оставлены на усмотрение тех, кто организует или реализует решение. В отсутствие конкретных профилей, регулирующих использование этого атрибута, он может использоваться, как подсказка для провайдеров идентификации о намерениях запрашивающей стороны сохранить идентификатор или связь его с местным значением.

Значение = "false" может использоваться для указания, что запрашивающая сторона не готова или не способна выполнить эти действия, и действия провайдера идентификации оказались бесполезными.

Запрашивающие стороны, которые не используют этот атрибут для конкретных целей, должны, как правило, установить его в значение "true" для максимизации возможностей взаимодействия. Атрибут `AllowCreate` не должен использоваться и должен быть проигнорирован вместе с запросами или подтверждениями, созданными с идентификаторами имен, имеющими `Format` со значением `urn:oasis:names:tc:SAML:2.0:nameid-format:transient` (они препятствуют возникновению для себя и внутри себя такого состояния).

Когда этот элемент используется, если его содержание не понятно или неприемлемо для провайдер идентификации, то элемент сообщения `<Response>` должен быть возвращен с указанием ошибки `<Status>`, и может содержать код состояния второго уровня `<StatusCode>`, имеющий значение `urn:oasis:names:tc:SAML:2.0:status:InvalidNameIDPolicy`.

Если `Format` пропущен или установлен в `urn:oasis:names:tc:SAML:2.0:nameid-format:unspecified`, то провайдер идентификации может возвращать идентификатор любого типа, учитывая все другие дополнительные ограничения из-за содержания этого элемента или правил провайдера идентификации или клиента.

Специальное значение `Format = urn:oasis:names:tc:SAML:2.0:nameid-format:encrypted` указывает, что результирующее(ие) подтверждение(я) должно(ы) вместо обычного текста содержать элементы `<EncryptedID>`. Зашифрованная форма базового идентификатора имени может быть любого типа, поддерживаемого провайдером идентификации для запрошенного объекта.

ПРИМЕЧАНИЕ 3 (информативное). – PE6 (см. OASIS PE:2006) предлагает добавить следующий текст в конец вышеприведенного параграфа:

Провайдер услуг не имеет возможности запрашивать, какой конкретно тип идентификатора должен быть возвращен, если он запрашивает шифрование. Элемент метаданных `<md:NameIDFormat>`, описанный в разделе 9 или иных документах, может использоваться для определения того, какой тип идентификатора должен быть зашифрован или возвращен.

ПРИМЕЧАНИЕ 4 (информативное). – PE15 (см. OASIS PE:2006) предлагает добавить следующий параграф:

Когда `Format`, определенный в 8.7.3.7 использует значения, не равные `urn:oasis:names:TC:SAML:2.0:nameid-format:unspecified` или `urn:oasis:names:TC:SAML:2.0:nameid-format:encrypted`, то, если провайдер идентификации возвращает одно из подтверждений:

- значение `Format = <NameID>` внутри `<Subject>` или любого `<Assertion>` должно быть идентично значению `Format`, представленному в `<NameIDPolicy>`; и
- если `SPNameQualifier` в `<NameIDPolicy>` не пропущено, то значение `SPNameQualifier` идентификатора `<NameID>` в `<Subject>` любого `<Assertion>` должно быть идентично значению `SPNameQualifier` , представленному в `<NameIDPolicy>`.

Вне зависимости от значения `Format` в `<NameIDPolicy>`, провайдер идентификации может вернуть в результирующем `<EncryptedID>` объекта подтверждения, действующие правила провайдера идентификации (возможно конкретные для данного провайдера услуг) требуют, чтобы использовался зашифрованный идентификатор.

Если запрашивающая сторона желает разрешить провайдеру идентификации создать новый идентификатор для клиента, если не существует ни одного, она должна включить этот элемент с атрибутом `AllowCreate` = "true". В противном случае успешно аутентифицирован может быть только клиент, для которого провайдер идентификации ранее установи идентификатор, который может использовать запрашивающая сторона. Это особенно полезно при использовании вместе со значением `Format` = `urn:oasis:names:tc:SAML:2.0:nameid-format:persistent` (см. раздел 12).

ПРИМЕЧАНИЕ 5 (информативное). – PE14 (см. OASIS PE:2006) предлагает игнорировать вышеприведенный параграф.

Приведенный далее фрагмент схемы определяет элемент `<NameIDPolicy>` и его сложный тип **NameIDPolicyType**:

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

#### 8.2.4.2 Элемент <Scoping>

Элемент `<Scoping>` определяет провайдеров идентификации, которым запрашивающая сторона доверяет аутентифицировать представителя, а также ограничения и контекст, связанные с ретрансляцией отвечающей стороной сообщения `<AuthnRequest>` последующим провайдерам идентификации. Его сложный тип **ScopingType** определяет следующие элементы и атрибут:

– `ProxyCount` [Дополнительный]

Определяет количество ретрансляций, допустимых между провайдером идентификации, который получает этот запрос `<AuthnRequest>` и провайдером идентификации, который, в конце концов, аутентифицирует клиента. Значение "ноль" не разрешает ретрансляций, если этот элемент пропущен, не налагается никаких ограничений.

– `<IDPList>` [Дополнительный]

Информативный список провайдеров идентификации и связанная с ними информация, которую запрашивающая сторона считает приемлемой для ответа на запрос.

– `<RequesterID>` [Ноль или несколько]

Идентифицирует множество запрашивающих элементов, от чьего имени действует запрашивающая сторона. Используется для связи в цепи запрашивающих сторон, когда используется ретрансляция, как описано в 8.2.4.5. Описание идентификаторов элемента содержится в 8.7.3.6.

В профилях, определяющих активный промежуточный элемент, промежуточный элемент может изучить этот список и вернуть сообщение `<Response>` с указанием ошибочного кода `<Status>` или кода второго уровня `<StatusCode>` = `urn:oasis:names:tc:SAML:2.0:status:NoAvailableIDP`, или `urn:oasis:names:tc:SAML:2.0:status:NoSupportedIDP`, если он не может связаться или не поддерживает любого из указанных провайдеров идентификации.

Приведенный далее фрагмент схемы определяет элемент `<Scoping>` и его сложный тип **ScopingType**:

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

#### 8.2.4.3 Элемент <IDPList>

Элемент `<IDPList>` определяет провайдеров идентификации, которым запрашивающая сторона доверяет аутентифицировать представителя. Его сложный тип **IDPListType** определяет следующие элементы:

– `<IDPEntry>` [Один или несколько]

Информация об одном-единственном провайдере идентификации.

– `<GetComplete>` [Дополнительный]

Если список `<IDPList>` не полон, использование этот элемент определяет ссылку на UR, который может использоваться для получения полного списка. Получение ресурсов, ассоциированных с URI, может привести к получению XML элемента, содержащего корневой элемент `<IDPList>`, который сам по себе не содержит элемента `<GetComplete>`.

Приведенный далее фрагмент схемы определяет элемент <IDPList> и его сложный тип **IDPListType**:

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

Элемент <IDPEntry> определяет одного-единственного провайдера идентификации, которому запрашивающая сторона доверяет аутентифицировать представителя. Его сложный тип **IDPEntryType** определяет следующий атрибуты:

– ProviderID [Требуемый]

Уникальный идентификатор провайдера идентификации. Описание таких идентификаторов содержится в 8.7.3.6.

– Name [Дополнительный]

понятное человеку название провайдера идентификации.

– Loc [Дополнительный]

Ссылка на URI, представляющий местоположение оконечной точки, определенной профилем, поддерживающим протокол запроса аутентификации. Связь, которая должна использоваться, должна быть понятна для применяемого профиля.

Приведенный далее фрагмент схемы определяет элемент <IDPEntry> и его сложный тип **IDPEntryType**:

```
<element name="IDPEntry" type="samlp:IDPEntryType"/>
<complexType name="IDPEntryType">
 <attribute name="ProviderID" type="anyURI" use="required"/>
 <attribute name="Name" type="string" use="optional"/>
 <attribute name="Loc" type="anyURI" use="optional"/>
</complexType>
```

#### 8.2.4.4 Правила обработки

Передача и прием сообщений <AuthnRequest> и <Response> может выполняться по самым разным сценариям и, следовательно, обычно определяется для использования в определенных условиях, в которых эти возможности ограничиваются и требуются или запрещены определенные виды входных и выходных данных. Приведенные ниже правила обработки применяются как неизменные для любого профиля данного протокола обмена данными. Также должны учитываться все остальные правила обработки, связанные с базовыми сообщениями запроса и ответа.

Отвечающая сторона должна в итоге ответить на запрос <AuthnRequest>, передав сообщение <Response>, содержащее одно или несколько подтверждений, которые соответствуют спецификации, определенной в запросе, или передав сообщение <Response>, содержащее элемент <Status>, описывающий возникшую ошибку. Отвечающая сторона может выполнить обмен сообщениями с представителем, если это требуется для инициации или завершения процесса аутентификации, в зависимости от природы связи протокола и механизма аутентификации. Как описано в следующем разделе, оно обмен включает в себя ретрансляцию запроса при помощи перенаправления представителя к другому провайдеру идентификации, создавая его собственное сообщение <AuthnRequest>, поэтому результирующее подтверждение может использоваться для аутентификации исходной отвечающей стороны, в действительности используя SAML в качестве механизма аутентификации.

Если отвечающая сторона не способна аутентифицировать представителя или не распознает запрошенный объект, или действующие правила провайдера идентификации запрещают ей создавать подтверждение (например, данный объект идентификации запрещает провайдеру идентификации передавать подтверждения данной доверяющей стороне), то он должен возвратить ответ <Response> с ошибкой <Status>, и может вернуть код второго уровня <StatusCode> = urn:oasis:names:tc:SAML:2.0:status:AuthnFailed; или urn:oasis:names:tc:SAML:2.0:status:UnknownPrincipal.

Если в запросе представлен элемент <saml:Subject>, то элемент <saml:Subject> результирующих подтверждений должен **точно соответствовать** элементу <saml:Subject> в запросе, как описано в 8.2.3.4, за исключением того, что идентификатор может иметь иной формат, если это указано в <NameIDPolicy>. В таком случае физическое содержание идентификатора может быть иным, но он должен описывать того же клиента.

Все содержание, специально определенное в запросе <AuthnRequest> является дополнительным, хотя некоторые из его компонентов могут требоваться различными профилями. При отсутствии какого-либо определенного содержания вообще, предполагается выполнение следующих действий:

- возвращаемое(ые) подтверждение(я) должно(ны) содержать элемент <saml:Subject>, который определяет представителя. Тип и фермат идентификатора определяются провайдером

идентификации. Как минимум в одном утверждении должно быть <saml:AuthnStatement>, которое описывает аутентификацию, выполняемую отвечающей стороной или связанной с ней службой аутентификации;

- представитель запроса должен, до возможной степени, быть одним-единственным аттестующим элементом, способным удовлетворять подтверждению(ям) <saml:SubjectConfirmation>. В случае применения боле слабых методов подтверждения для того чтобы обеспечить выполнение этого требования будут использоваться механизмы связей протокола или иные механизмы;
- результирующее(ие) подтверждение(я) должно(ы) содержать элемент <saml:AudienceRestriction>, указывающий запрашивающую сторону, как приемлемую доверяющую сторону. Если провайдер идентификации считает это приемлемым, могут быть включены и другие типы аудитории.

#### 8.2.4.5 Передача через промежуточный элемент

Если провайдер идентификации, который получает запрос <AuthnRequest>, еще не аутентифицировал представителя и не может аутентифицировать его непосредственно, но верит, что этот представитель уже аутентифицирован для другого провайдера идентификации или для эквивалентного не-SAML механизма, он может ответить на этот запрос, создав новый запрос <AuthnRequest> от своего имени и представив его другому провайдеру идентификации или запрос в любом не-SAML формате, который понимает элемент. Тот исходный провайдер идентификации называется промежуточным провайдером идентификации.

После успешного возвращения промежуточному провайдеру сообщения <Response> (либо его не-SAML эквивалента), вложенное в сообщение подтверждение либо его не-SAML эквивалент может использоваться для аутентификации представителя, так что этот промежуточный провайдер может сам создать подтверждение в ответ исходный запрос <AuthnRequest>, завершая общий процесс обмена сообщениями. Оба – промежуточный и аутентифицирующий провайдеры идентификации – могут накладывать ограничения на действия по ретрансляции в сообщениях и подтверждениях, которые они создают, как описано в предыдущих подразделах и далее.

Запрашивающая сторона может повлиять на действие промежуточного элемента, включая элемент <Scoping>, в котором провайдер помещает желаемое значение ProxyCount и/или приводит список предпочтительных провайдеров идентификации, которым можно ретранслировать сообщения при помощи включения упорядоченного списка предпочтительных провайдеров <IDPList>.

Провайдер идентификации может контролировать вторичное использование его подтверждения промежуточными провайдерами идентификации, используя в создаваемых им подтверждениях элемент <ProxyRestriction>.

Провайдер идентификации может ретранслировать запрос <AuthnRequest>, если атрибут <ProxyCount> пропущен или больше нуля. Выбирает ли он возможность ретрансляции или нет, зависит от местных правил работы. Провайдер идентификации может решить ретранслировать сообщение провайдеру, указанному в списке <IDPList>, если этот список представлен, но он не обязан этого делать.

Провайдер идентификации не должен ретранслировать запрос, в котором <ProxyCount> = 0. Провайдер идентификации должен возвратить сообщение об ошибке <Status>, содержащий код второго уровня <StatusCode> со значением urn:oasis:names:tc:SAML:2.0:status:ProxyCountExceeded, если только он не может непосредственно аутентифицировать представителя.

Если он решает ретранслировать сообщение провайдеру идентификации SAML, то, при создании нового запроса <AuthnRequest>, промежуточный провайдер идентификации должен включить в него эквивалентные или боле строгие формы всех данных, содержащихся в исходном запросе (такие как сведения о правилах контекста аутентификации). Отметим, однако, что промежуточный провайдер может указать любой элемент <NameIDPolicy>, если он стремится максимизировать вероятность положительного ответа.

Если аутентифицирующий провайдер идентификации не является SAML провайдером идентификации, то промежуточный провайдер должен иметь другой способ гарантировать, что элементы, обеспечивающие взаимодействие агента пользователя (например, <IsPassive>) будут приемлемыми для аутентифицирующего провайдера.

Новый запрос <AuthnRequest> должен содержать атрибут <ProxyCount>, значение которого не превышает числа, на единицу меньше, чем исходное значение этого атрибута. Если исходный запрос не содержит атрибута <ProxyCount>, то новый запрос должен содержать атрибут <ProxyCount>.

Если в исходном запросе был указан список <IDPList>, новый запрос также должен содержать список <IDPList>. Промежуточный провайдер идентификации может добавить дополнительных провайдеров идентификации в конец списка <IDPList>, но не должен ни одного из них удалять из списка.

Запрос аутентификации и ответ на него обрабатываются в обычном режиме в соответствии с правилами, определенными в настоящем разделе, и в соответствии с используемым профилем. После того как представитель аутентифицирован для промежуточного провайдера идентификации (в случае варианта SAML – путем доставки сообщения <Response>), выполняются следующие действия:

- промежуточный провайдер идентификации готовит новое подтверждение от своего собственного имени, копируя соответствующую информацию из исходного подтверждения либо его не-SAML эквивалента;
- элемент <saml:Subject> нового подтверждения должен содержать идентификатор, который соответствует предпочтениям исходной запрашивающей стороны, как определено в элементе <NameIDPolicy>;

- утверждение <saml:AuthnStatement> в новом подтверждении должно содержать элемент <saml:AuthnContext>, содержащий элемент <saml:AuthenticatingAuthority>, указывающий провайдера идентификации, к которому направляет представителя промежуточный провайдер идентификации. Если исходное подтверждение содержит информацию <saml:AuthnContext>, которая включает в себя один или несколько элементов <saml:AuthenticatingAuthority>, то эти элементы должны быть указаны в новом подтверждении, а после них должен быть помещен новый элемент;
- если аутентифицирующий провайдер идентификации не является провайдером SAML, то промежуточный провайдер идентификации должен создать для аутентифицирующего провайдера уникальное значение идентификатора. Это значение должно быть совместимо для различных запросов в течение определенного времени. Это значение не должно конфликтовать со значениями, используемыми или создаваемыми другими провайдерами SAML;
- любая другая информация <saml:AuthnContext> может быть скопирована, транслирована или пропущена в соответствии с правилами работы промежуточного провайдера идентификации, при условии, что выполняются исходные требования, определенные запрашивающей стороной.

Если в будущем провайдера идентификации попросят аутентифицировать того же самого представителя для второй запрашивающей стороны и этот запрос будет иметь тот же или более низкий уровень строгости, как у исходного запроса (как определено промежуточным провайдером идентификации), провайдер идентификации может не создавать новый запрос <AuthnRequest> для аутентифицирующего провайдера идентификации, а сразу же передать еще одно подтверждение (при условии, что исходное подтверждение либо его не-SAML эквивалент, который им был получен, все еще действует).

### 8.2.5 Протокол применения артефактов

Протокол разрешения артефактов обеспечивает механизм, при помощи которого Протокольные сообщения SAML могут быть транспортированы в связь SAML посредством ссылки, а не посредством передачи значения. Используя этот специальный протокол, и запросы и ответы могут быть получены по ссылке. Источник сообщения, вместо того, чтобы вводить сообщение в транспортный протокол, передает небольшой блок данных, называемый артефактом, используя связь протокола. Артефакт может иметь различную форму, но должен поддерживать средства, при помощи которых получатель может определить, кто передал его. Если получатель пожелает, он может затем использовать этот протокол вместе с различными (обычно синхронными) связями протокола SAML для применения этого артефакта в исходном протокольном сообщении.

Наиболее широко этот механизм применяется – со связями, которые не могут простым способом передать сообщение из-за ограничений по размеру, или потому что это сообщение не может быть передано по безопасному каналу между запрашивающей и отвечающей сторонами SAML, без необходимости подписи.

В зависимости от характеристики исходного сообщения, передаваемого посредством ссылки, протокол применения артефактов может потребовать от используемой связи протокола обеспечить защиту, например, взаимной аутентификации, защиты целостности, конфиденциальности и т. д., для применения данного артефакта. Во всех случаях, этот артефакт должен проявлять семантику одноразового использования, так как после того как он успешно применен, он более не может использоваться какой-либо стороной.

Вне зависимости от полученного сообщения протокола, результат применения артефакта должен обрабатываться точно так же, как, если бы вместо артефакта было бы получено исходное сообщение.

#### 8.2.5.1 Элемент <ArtifactResolve>

Сообщение <ArtifactResolve> используется для запроса того, чтобы сообщение протокола SAML было возвращено в сообщении <ArtifactResponse>, путем указания артефакта, который представляет сообщение протокола SAML. Исходная передача артефакта регулируется конкретной связью протокола, который используется.

Сообщение <ArtifactResolve> должно быть подписано или в противном случае аутентифицировано, его целостность должна быть защищена связью протокола, используемого для доставки сообщения.

Это сообщение имеет сложный тип **ArtifactResolveType**, который расширяет **RequestAbstractType** и добавляет следующий элемент:

– <Artifact> [Требуемый]

Значение артефакта, который получила запрашивающая сторона, и теперь желает преобразовать его в сообщение протокола, который она представляет.

Приведенный далее фрагмент схемы определяет элемент <ArtifactResolve> и его сложный тип **ArtifactResolveType**:

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

#### 8.2.5.2 Элемент <ArtifactResponse>

Получатель сообщения <ArtifactResolve> должен ответить, передав элемент сообщения <ArtifactResponse>. Этот элемент имеет сложный тип **ArtifactResponseType**, который расширяет **StatusResponseType**, добавляя один-единственный дополнительный подстановочный элемент, соответствующий возвращаемому сообщению протокола SAML. Этот свернутый элемент сообщения может быть и запросом, и ответом.

Сообщение <ArtifactResponse> должно быть подписано или в противном случае аутентифицировано, его целостность должна быть защищена связью протокола, используемого для доставки сообщения.

Приведенный далее фрагмент схемы определяет элемент <ArtifactResponse> и его сложный тип **ArtifactResponseType**:

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

#### 8.2.5.3 Правила обработки

Если отвечающая сторона распознает артефакт как достоверный, то она отвечает, передавая соответствующее сообщение протокола в элементе сообщения <ArtifactResponse>. В противном случае она отвечает, передавая элемент <ArtifactResponse>, не содержащий вложенных сообщений. В обоих случаях элемент <Status> должен содержать элемент <StatusCode> со значением кода urn:oasis:names:tc:SAML:2.0:status:Success. Сообщение-ответ, не содержащее внутри себя вложенных сообщений, называется пустым ответом в оставшейся части данного раздела.

Отвечающая сторона должна наложить на данный артефакт требование одноразового использования, обеспечивая, что любой последующий запрос с тем же артефактом от любой запрашивающей стороны приведет к созданию пустого ответа, как описано выше.

Некоторые протокольные сообщения SAML, точнее, сообщение <AuthnRequest> в некоторых профилях, могут быть предназначены для использования любой стороной, которая его получает и может соответствующим образом ответить. В большинстве других случаев, однако, сообщение предназначено для конкретного элемента. В таких ситуациях, артефакт, когда он создан, должен быть ассоциирован с конкретным получателем сообщения, в котором содержится этот артефакт. Если создатель артефакта получает сообщение <ArtifactResolve> от запрашивающей стороны, которая не может аутентифицировать себя, как исходный получатель, которому предназначен этот артефакт, то создатель артефакта должен возвратить пустой ответ.

Создатель артефакта должен установить минимально возможный предел времени использования артефакта, например, существует допустимый интервал времени (но не более), в течение которого получатель артефакта может получить артефакт и возвратить его создателю в сообщении <ArtifactResolve>.

Атрибут InResponseTo сообщения <ArtifactResponse> должен содержать значение соответствующего атрибута ID сообщения <ArtifactResolve>, но вложенное сообщение протокола будет содержать свой собственный идентификатор сообщения, и в случае наличия вложенного ответа, может содержать иное значение InResponseTo, которое соответствует исходному сообщению запроса, на которое отвечает вложенное сообщение.

Необходимо учитывать все другие правила обработки, связанные с базовыми сообщениями запроса и ответа.

### 8.2.6 Протокол управления идентификатором имени

После назначения идентификатора имени клиента, провайдер идентификации, желающий изменить значение и/или формат идентификатора, который он будет использовать для обозначения этого клиента, или указать, что этот идентификатор имени более не указывает этого клиента, сообщает провайдерам услуг об изменении, передавая им сообщение `<ManageNameIDRequest>`.

ПРИМЕЧАНИЕ 1 (информативное). – PE12 (см. OASIS PE:2006) идентифицирует назначение вышеприведенного параграфа, переписывая его следующим образом:

> После назначения идентификатора имени клиента, провайдер идентификации, желающий изменить значение идентификатора, который он будет использовать для обозначения этого клиента, или указать, что этот идентификатор имени более не указывает этого клиента, сообщает провайдерам услуг об изменении, передавая им сообщение `<ManageNameIDRequest>`.

Провайдер услуг также использует это сообщение для записи или изменения значения `SPProvidedID`, которое должно быть включено, когда для связи с ним используется базовый идентификатор имени, или для завершения использования идентификатора имени при общении между ним и провайдером идентификации.

Этот протокол, как правило, не используется с "временными" идентификаторами имен, поскольку их значения не предназначены для использования в течение длительного времени.

ПРИМЕЧАНИЕ 2 (информативное). – PE14 (см. OASIS PE:2006) разъясняет этот текст следующим образом:

> Этот протокол не должен использоваться вместе с форматом `urn:oasis:names:tc:SAML:2.0:nameidformat:transient <NameID> Format`.

#### 8.2.6.1 Элемент <ManageNameIDRequest>

Провайдер передает сообщение `<ManageNameIDRequest>` для того, чтобы сообщить получателю об изменении идентификатора имени или сообщить о прекращении использования идентификатора имени.

Сообщение `<ManageNameIDRequest>` должно быть подписано или в противном случае аутентифицировано, его целостность должна быть защищена связью протокола, используемого для доставки сообщения.

Это сообщение имеет сложный тип **ManageNameIDRequestType**, который расширяет **RequestAbstractType** и добавляет следующие элементы:

– `<saml:NameID>` или `<saml:EncryptedID>` [Требуемый]

  Идентификатор имени и связанные с ним данные описания (в виде обычного текста или в зашифрованном виде), которые определяют клиента, как распознанного на данный момент его провайдером идентификации и провайдером услуг до передачи данного запроса (более подробная информация об этих элементах приведена в 8.1.2).

– `<NewID>` или `<NewEncryptedID>` или `<Terminate>` [Требуемый]

  Новое значение идентификатора (в виде обычного текста или в зашифрованном виде), которое должно использоваться при общении с запрашивающим провайдером относительно данного клиента, или для указания, что использование старого идентификатора закончено. В первом случае, если запрашивающей стороной является провайдер услуг, новый идентификатор должен появиться в элементах `<NameID>` атрибута `SPProvidedID`. Если запрашивающей появится в последующих элементах `<NameID>` в виде содержания элемента.

  ПРИМЕЧАНИЕ (информативное). – PE12 (см. OASIS PE:2006) предлагает дополнить вышеприведенный параграф следующим текстом:

  В любом случае если используется `<NewEncryptedID>`, то его зашифрованное значение – это просто элемент `<NewID>`, содержащий только новое значение идентификатора (формат и определители после их определения меняться не могут).

Приведенный далее фрагмент схемы определяет элемент `<ManageNameIDRequest>` и его сложный тип **ManageNameIDRequestType**:

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

```
<element name="NewID" type="string"/>
<element name="NewEncryptedID" type="saml:EncryptedElementType"/>
<element name="Terminate" type="samlp:TerminateType"/>
<complexType name="TerminateType"/>
```

#### 8.2.6.2 Элемент <ManageNameIDResponse>

Получатель сообщения <ManageNameIDRequest> должен ответить, передавая сообщение <ManageNameIDResponse>, имеющее тип **StatusResponseType**, без дополнительного содержания.

Сообщение <ManageNameIDResponse> должно быть подписано или в противном случае аутентифицировано, его целостность должна быть защищена связью протокола, используемого для доставки сообщения.

Приведенный далее фрагмент схемы определяет элемент <ManageNameIDResponse>:

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

#### 8.2.6.3 Правила обработки

Если запрос содержит <saml:NameID> (или его зашифрованную версию), которую получатель не может распознать, отвечающий провайдер должен ответить, передавая сообщение ошибки <Status>, и может ответить, передавая код второго уровня <StatusCode> со значением urn:oasis:names:tc:SAML:2.0:status:UnknownPrincipal.

ПРИМЕЧАНИЕ 1 (информативное). – PE14 (см. OASIS PE:2006) далее разъясняет этот параграф ниже. Более подробно см. Дополнение VIII.

Если в запрос включен элемент <Terminate, то представляющий его провайдер указывает, что (в случае для провайдера услуг) он больше не будет принимать подтверждения от этого провайдера идентификации или (в случае для провайдера идентификации) он больше не будет создавать подтверждения об этом клиенте для этого провайдера услуг. Принимающий провайдер может выполнять любую поддержку, не имея понятия о прекращении действия взаимосвязи между клиентом идентификатором имени. Он может пожелать прекратить активный(е) сеанс(ы) связи клиента, для которого прекратилось действие взаимосвязи.

ПРИМЕЧАНИЕ 2 (информативное). – PE8 (см. OASIS PE:2006) предлагает заменить последнее предложение этого параграфа следующим текстом:

Как правило, не следует прерывать любой(ые) активный(е) сеанс(ы) связи клиента, для которого прекратилось действие взаимосвязи. Если принимающим провайдером является провайдер идентификации, он не должен прерывать никакого(их) активный(е) сеанс(ы) связи клиент, установленного(ых) с другими провайдерами услуг. Принимающий провайдер может передать сообщение <LogoutRequest> завершения действия идентификатора имени, передав сообщение <ManageNameIDRequest>, если это является намерением принимающего провайдера (например, прекращение действия идентификатора имени инициировано при помощи администратора, который хотел завершить все действия пользователя). Принимающий провайдер не должен передавать сообщение <LogoutRequest> после того, как передано сообщение <ManageNameIDRequest>.

Если провайдер услуг запрашивает изменение своего идентификатора клиента путем включения элемента <NewID> (или <NewEncryptedID>), провайдер идентификации должен включать содержание элемента в виде SPProvidedID , когда после этого обменивается с провайдером услуг информацией об этом клиенте.

Если провайдер идентификации запрашивает изменение своего идентификатора клиента путем включения элемента <NewID> (или <NewEncryptedID>), провайдер услуг должен использовать содержание элемента в виде содержание элемента <saml:NameID> когда после этого обменивается с провайдером идентификации информацией об этом клиенте.

Ни один из этих аутентификаторов, один из них или оба – исходный и новый может быть зашифрован (с использованием элементов <EncryptedID> и <NewEncryptedID>).

В любом случае содержание <saml:NameID> в запросе и связанном с ним атрибуте SPProvidedID должно содержать наиболее последнюю информацию идентификатора имени, о котором достигнуто соглашение между провайдерами относительно данного клиента.

В том случае, когда имеется идентификатор с форматом Format = urn:oasis:names:tc:SAML:2.0:nameid-format:persistent, атрибут NameQualifier должен содержать уникальный идентификатор провайдера идентификации, который создал этот идентификатор. Если идентификатор был согласован между провайдером идентификации и группой, в которую входит этот провайдер услуг, то атрибут SPNameQualifier должен содержать уникальный идентификатор этой группы. В противном случае он должен содержать уникальный идентификатор провайдера услуг. Эти атрибуты могут быть пропущены, если в противном случае они будут соответствовать значении, содержащему элемент <Issuer> сообщения прокола, но это не рекомендуется из-за возможности недоразумений.

Для изменения этих идентификаторов может потребоваться огромных затрат времени как запрашивающей, так и отвечающей стороны для распространения информации о них в системе. Разработчики могут пожелать разрешить каждой стороне принять любой идентификатор на некоторый период времени с последующим успешным завершением смены идентификатора имени. Не выполнение этого может привести к невозможности клиента получить доступ к ресурсам.

Должны учитываться все остальные правила обработки, связанные с базовыми сообщениями запроса и ответа.

### 8.2.7 Протокол единого выхода из системы

Протокол единого выхода из системы представляет собой протокол обмена сообщениями, при помощи которого все сеансы связи, поддерживаемые определенным ответственным органом сеанса связи, почти одновременно завершаются. Протокол единого выхода из системы используется либо, когда клиент отключается на стороне участника сеанса связи, или когда клиент отключается непосредственно на стороне ответственного органа сеанса связи. Этот протокол может также использоваться для отключения клиента по истечении времени. Причина события отключения может быть указана в атрибуте `Reason`.

Клиент может иметь установленные аутентифицированные сеансы связи как с ответственным органом сеанса связи, так и с отдельными участниками сеанса связи, базирующиеся на подтверждениях, содержащих утверждения аутентификации, предоставленные ответственным органом сеанса связи.

Когда клиент запускает процесс единого выхода из системы на стороне участника сеанса связи, этот участник сеанса связи должен передать сообщение `<LogoutRequest>` ответственному органу сеанса связи, которым были предоставлено подтверждение, содержащее утверждение аутентификации, относящееся к этому сеансу связи на стороне участника сеанса связи.

Когда либо клиент запускает процесс единого выхода из системы на стороне ответственного органа сеанса связи, либо участник сеанса связи передает запрос на выход из системы ответственному органу сеанса связи, определяющему этого клиента, ответственный орган сеанса связи должен передать сообщение `<LogoutRequest>` каждому участнику сеанса связи, которому он посылал подтверждения, содержащие утверждения аутентификации во время текущего сеанса связи с этим клиентом, за исключением того участника сеанса связи, который передал ответственному органу сеанса связи сообщение `<LogoutRequest>`. Он должен попытаться связаться с максимально возможным числом этих участников сеанса связи, которое позволит данный протокол, завершить свой собственный сеанс связи с клиентом и, наконец, вернуть сообщение `<LogoutResponse>` запрашивающему участнику сеанса связи, если таковой имеется.

#### 8.2.7.1 Элемент <LogoutRequest>

Участник сеанса связи или ответственный орган сеанса связи передает сообщение `<LogoutRequest>` для указания того, что сеанс связи закончен.

Сообщение `<LogoutRequest>` должно быть подписано или в противном случае аутентифицировано, его целостность должна быть защищена связью протокола, используемого для доставки сообщения.

Это сообщение имеет сложный тип **LogoutRequestType**, который расширяет **RequestAbstractType** и добавляет следующие элементы и атрибуты:

– `NotOnOrAfter` [Дополнительный]

Время, истечения срока действия запроса, после которого получатель может отбросить сообщение. Значение времени кодируется в единицах UTC, как описано в 7.3.

– `Reason` [Дополнительный]

Указание причины выхода из системы, в виде ссылки на URI.

ПРИМЕЧАНИЕ 1 (информативное). – PE10 (см. OASIS PE:2006) предлагает заменить вышеприведенный текст следующим:

Атрибут: `Reason` определен в схеме как строка. Такое определение еще больше ограничивает схему, требуя, чтобы атрибут `Reason` имел форму ссылки на URI.

– `<saml:BaseID>` *или* `<saml:NameID>` *или* `<saml:EncryptedID>` [Требуемый]

Идентификатор и связанные с ним атрибуты (в виде обычного текста или в зашифрованном виде) которые определяют клиента, как распознанного на данный момент его провайдером идентификации и провайдером услуг до передачи данного запроса (Более подробная информация об этих элементах приведена в 8.1.2).

– `<SessionIndex>` [Дополнительный]

Идентификатор, который присваивает номер этому сеансу связи на стороне получателя.

ПРИМЕЧАНИЕ 2 (информативное). – PE38 (см. OASIS PE:2006) разъясняет этот текст следующим образом:

Номер сеанса связи между клиентом, определенным элементом `<saml:BaseID>`, `<saml:NameID>` или `<saml:EncryptedID>`, и ответственным органом сеанса связи. Он должен быть коррелирован с атрибутом `SessionIndex`, если таковой имеется, в элементе `<saml:AuthnStatement>` подтверждения, использованного для установления сеанса связи, завершение которого выполняется.

Приведенный далее фрагмент схемы определяет элемент <LogoutRequest> и связанный с ним сложный тип **LogoutRequestType**:

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

#### 8.2.7.2 Элемент <LogoutResponse>

Получатель сообщения <LogoutRequest> должен ответить, передав сообщение <LogoutResponse> типа **StatusResponseType**, без указания дополнительного содержания.

Сообщение <LogoutResponse> должно быть подписано или в противном случае аутентифицировано, его целостность должна быть защищена связью протокола, используемого для доставки сообщения.

Приведенный далее фрагмент схемы определяет элемент <LogoutResponse>:

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

#### 8.2.7.3 Правила обработки

Отправитель сообщения может использовать атрибут Reason для указания причины передачи запроса <LogoutRequest>. следующие значения определяются в настоящей Рекомендации для использования всеми отправителями сообщений; другие значения могут быть согласованы между участниками:

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

Определяет, что сообщение передается, потому что клиент желает завершить указанный сеанс связи.

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

Определяет, что сообщение передается, потому что администратор желает завершить указанный сеанс связи для данного клиента.

Должны учитываться все остальные правила обработки, связанные с базовыми сообщениями запроса и ответа.

Дополнительные правила обработки приводятся в последующих подразделах.

**1) Правила для участника сеанса связи**

Когда участник сеанса связи получает сообщение <LogoutRequest>, участник сеанса связи должен аутентифицировать сообщение. Если отправителем является ответственный орган, который предоставляет подтверждение, содержащее утверждение аутентификации, связанное с текущим сеансом связи клиента, то участник сеанса связи должен признать недействительным(и) и сеанс(ы) связи клиента, обозначенные элементом <saml:BaseID>, <saml:NameID> или <saml:EncryptedID>, и любые элементы <SessionIndex>, представленные в этом сообщении. Если не представлено ни одного элемента <SessionIndex>, то все сеансы связи, связанные с этим клиентом, должны быть признаны недействительными.

Участник сеанса связи должен применять сообщение запроса выхода из системы к любому подтверждению, которое удовлетворяет следующим условиям, даже если подтверждение приходит позже, чем запрос выхода из системы:

- Объект подтверждения **точно соответствует** элементу <saml:BaseID>, <saml:NameID> или <saml:EncryptedID> в запросе <LogoutRequest>, как определено в 8.2.3.4.

- Атрибут SessionIndex одного из утверждений подтверждения аутентификации соответствует одному из двух элементов <SessionIndex>, указанных в запросе выхода из системы, либо запрос выхода из системы не содержит элементов <SessionIndex>.
- В противном случае подтверждение будет достоверным, на основании условий времени, указанных в самом подтверждении (в частности, значение для данных условий любых определенных атрибутов NotOnOrAfter определяется данными подтверждения).

Еще не истек срок существования запроса выхода из системы (определяется при помощи рассмотрения атрибута NotOnOrAfter для этого сообщения).

ПРИМЕЧАНИЕ. – Это правило предназначено для предотвращения такой ситуации, когда участник сеанса связи принимает запрос выхода из системы, направленный на одно или несколько подтверждений, указанных элементом(амии) <SessionIndex>, *до того как* он получит текущее и, возможно, все еще действующее подтверждения, на которое направлен запрос выхода из системы. Он должен поддерживать запрос выхода из системы до тех пор, пока не может быть отброшен сам запрос выхода из системы (превышено значение NotOnOrAfter в запросе) или не будет получено и соответствующим образом обработано подтверждение, направленное запросом выхода из системы.

**2) Правила для ответственного органа сеанса связи**

Когда ответственный орган сеанса связи получает сообщение <LogoutRequest>, ответственный орган сеанса связи должен аутентифицировать отправителя. Если отправителем является участник сеанса связи, для которого ответственный орган сеанса связи предоставляет подтверждение, содержащее утверждение аутентификации для текущего сеанса связи, то ответственный орган сеанса связи должен выполнить следующие действия в указанном порядке:

– передать сообщение <LogoutRequest> любому ответственному органу сеанса связи, от лица которого данный ответственный орган сеанса связи ретранслирует данные аутентификации клиента, если только этот второй ответственный орган не является создателем запроса <LogoutRequest>;

– передать сообщение <LogoutRequest> каждому участнику сеанса связи, для которых ответственный орган сеанса связи предоставляет подтверждения в ходе текущего сеанса связи, *не являющимся* создателем текущего запроса <LogoutRequest>;

– завершить сеанс связи клиента, как определено элементом <saml:BaseID>, <saml:NameID> или <saml:EncryptedID>, и любыми элементами <SessionIndex>, представленными в сообщении запроса выхода из системы.

Если ответственный орган сеанса связи успешно завершает сеанс связи клиента по отношению к себе, то он должен ответить исходной запрашивающей стороне, если таковая имеется, передав сообщение <LogoutResponse>, содержащее код состояния высшего уровня со значением urn:oasis:names:tc:SAML:2.0:status:Success. Если он не может сделать этого, то он должен ответить, передавая сообщение <LogoutResponse>, содержащее код состояния высшего уровня, указывающий ошибку. Таким образом, состояние высшего уровня указывает состояние операции выхода из системы только по отношению к самому ответственному органу сеанса связи.

Ответственный орган сеанса связи попытается связаться с каждым участником сеанса связи, используя любую применимую/подходящую связь протокола, даже если одна или несколько этих попыток окажутся неудачными или не могут быть выполнены (например, потому что исходный запрос выполняется с использованием связи протокола, которая не может распространить команду выхода из системы на всех участников).

В том случае, когда не все участники сеанса связи успешно отвечают на сообщение <LogoutRequest> (или если не со всеми участниками можно установить контакт), то ответственный орган сеанса связи должен содержать в своем сообщении <LogoutResponse> код состояния второго уровня со значением urn:oasis:names:tc:SAML:2.0:status:PartialLogout для указания, что не все другие участники сеанса связи ответили и подтвердили выход из системы.

Ответственный орган сеанса связи может инициировать выход из системы по причинам, отличным от получения запроса <LogoutRequest> от участника сеанса связи – эти причины включают в себя, но не ограничиваются перечисленным:

- если с отдельным участником сеанса связи был согласован некоторый период ожидания, то ответственный орган сеанса связи может передать запрос <LogoutRequest> только этому отдельному участнику;
- превышено глобально согласованное время ожидания;
- клиент или другая доверенная сторона запросили вывести данного клиента из системы непосредственно на стороне ответственного органа сеанса связи;
- ответственный орган сеанса связи определил, что данные, подтверждающие личность клиента недостоверны.

При создании сообщения запроса выхода из системы, ответственный орган сеанса связи должен установит значение атрибута NotOnOrAfter сообщения в значение времени, указывающее время истечения для сообщения, после которого получатель может отбросить запрос выхода из системы. Эта величина должна иметь значение времени, которое равное или большее значения любого атрибута NotOnOrAfter, указанного в подтверждении, созданного последним, как часть целевого сеанса связи (как указано в атрибуте SessionIndex запроса выхода из системы).

В дополнение к значениям, определенным в 8.2.6.3 для атрибута Reason, доступны также следующие значения, которые могут применяться только ответственным органом сеанса связи:

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

Определяет, что сообщение передается потому, что превышен интервал времени глобального сеанса связи.

urn:oasis:names:tc:SAML:2.0:logout:sp-timeout

Определяет, что сообщение передается, потому что превышен период ожидания, согласованный между участником сеанса связи и ответственным органом сеанса связи.

### 8.2.8 Протокол преобразования идентификатора имени

Когда элемент, который использует идентификатор клиента совместно с провайдером идентификации, желает получить идентификатор имени для того же клиента в определенном формате или в объединенной области имен, он может передать запрос провайдеру идентификации, используя этот протокол.

Например, провайдер услуг, желающий установить связь с другим провайдером услуг, с которым он не использует совместно идентификатор клиента, может использовать провайдера идентификации, который использует идентификатор клиента совместно с обоими провайдерами услуг, для преобразования своего собственного идентификатора в новый идентификатор, обычно зашифрованный, при помощи которого он может установить связь со вторым провайдером услуг.

Вне зависимости от типа используемого идентификатора, преобразуемый идентификатор должен быть зашифрован и передан в элемент <saml:EncryptedID>, если только особые условия использования не указывают на то, что этот протокол не нужен.

#### 8.2.8.1 Элемент <NameIDMappingRequest>

Для того чтобы запросить у провайдера идентификации другой идентификатор имени для клиента, запрашивающая сторона передает сообщение <NameIDMappingRequest>. Это сообщение имеет сложный тип **NameIDMappingRequestType**, который расширяет **RequestAbstractType** и добавляет следующие элементы:

– <saml:BaseID> или <saml:NameID> или <saml:EncryptedID> [Требуемый]

  Идентификатор и связанная с ним описательная информация, которые определяют клиента, как полностью распознанного как запрашивающей, так и отвечающей сторон. (Более подробная информация об этом элементе представлена в 8.1.2.)

– <NameIDPolicy> [Требуемый]

  Требования относительно формата и дополнительного спецификатора имени для идентификатора, который должен быть возвращен.

  Сообщение должно быть подписано или в противном случае аутентифицировано, его целостность должна быть защищена связью протокола, используемого для доставки сообщения.

Приведенный далее фрагмент схемы определяет элемент <NameIDMappingRequest> и его сложный тип **NameIDMappingRequestType**:

```
<element name="NameIDMappingRequest" type="samlp:NameIDMappingRequestType"/>
<complexType name="NameIDMappingRequestType">
 <complexContent>
   <extension base="samlp:RequestAbstractType">
     <sequence>
       <choice>
         <element ref="saml:BaseID"/>
         <element ref="saml:NameID"/>
         <element ref="saml:EncryptedID"/>
       </choice>
       <element ref="samlp:NameIDPolicy"/>
     </sequence>
   </extension>
 </complexContent>
</complexType>
```

#### 8.2.8.2 Элемент <NameIDMappingResponse>

Получатель сообщения <NameIDMappingRequest> должен ответить, передавая сообщение <NameIDMappingResponse>. Это сообщение имеет сложный тип **NameIDMappingResponseType**, который расширяет **StatusResponseType** и добавляет следующий элемент:

– <saml:NameID> или <saml:EncryptedID> [Требуемый]

  Идентификатор и связанные с ним атрибуты, которые определяют клиента требуемым образом, обычно представлены в зашифрованном виде. (Более подробная информация об этом элементе представлена в 8.1.2.)

Сообщение должно быть подписано или в противном случае аутентифицировано, его целостность должна быть защищена связью протокола, используемого для доставки сообщения.

Приведенный далее фрагмент схемы определяет элемент <NameIDMappingResponse> и его сложный тип **NameIDMappingResponseType**:

```
<element name="NameIDMappingResponse"
type="samlp:NameIDMappingResponseType"/>
<complexType name="NameIDMappingResponseType">
 <complexContent>
    <extension base="samlp:StatusResponseType">
       <choice>
          <element ref="saml:NameID"/>
          <element ref="saml:EncryptedID"/>
       </choice>
    </extension>
 </complexContent>
</complexType>
```

#### 8.2.8.3 Правила обработки

Если отвечающая сторона не распознает клиента, указанного в запросе, она может ответить, передав сообщение об ошибке <Status> содержащий код второго уровня <StatusCode>, имеющий значение:

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

На усмотрение отвечающей стороны может быть возвращен код состояния urn:oasis:names:tc:SAML:2.0:status:InvalidNameIDPolicy, который указывает невозможность или нежелание передать идентификатор в запрошенном формате или области имен.

Должны учитываться все остальные правила обработки, связанные с базовыми сообщениями запроса и ответа.

### 8.3 Контроль версий SAML

Комплект Рекомендаций по SAML развивается в двух независимых направлениях. Каждое из них рассматривается в последующих подразделах вместе с правилами обработки для определения и обработки различий в версиях. Кроме того, добавлены рекомендации о том, когда и почему ожидается изменение данных конкретной версии в будущем при пересмотре SAML.

Когда информация о версии выражена как две версии – Основная и Дополнительная, она выражена в форме *Major.Minor*. Номер версии $Major_B.Minor_B$ выше, чем номер версии $Major_A.Minor_A$, если и только если:

$$(\text{Major}_B > \text{Major}_A) \text{ ИЛИ } ((\text{Major}_B = \text{Major}_A) \text{ И } (\text{Minor}_B > \text{Minor}_A)).$$

#### 8.3.1 Версия комплекта спецификаций языка SAML

Каждая версия Рекомендации по SAML будет содержать обозначения основной и дополнительной версий, описывающие их взаимосвязь с более ранними и более поздними версиями настоящей Рекомендации. Версия будет выражена в содержании настоящей Рекомендации. Общий размер и область изменений настоящей Рекомендации будет неофициально определять, образует ли множество изменений крупный или мелкий пересмотр. Как правило, общие изменения остаются совместимыми назад (с более ранними вариантами), тогда новая версия будет представлять собой незначительные изменения. В противном случае изменения будут приводить к крупному пересмотру.

Настоящая Рекомендация определяет версию 2.0.

##### 8.3.1.1 Версия схемы

Не являясь нормативным механизмом, любая документация XML схемы, опубликованная в виде части набора спецификаций, будет содержать атрибут версии в элементе <xs:schema>, значение которого имеет вид *Major.Minor*, показывая версию набора спецификаций, в которой она опубликована. В утвержденных вариантах реализации может использоваться атрибут в качестве средства указания на то, какая версия схемы используется для проверки достоверности сообщения, или для поддержания нескольких версий одной логической схемы.

##### 8.3.1.2 Версия подтверждения SAML

Элемент языка SAML <Assertion> содержит атрибут для выражения основной или дополнительной версии подтверждения в строчной форме *Major.Minor*. Каждая версия набора спецификаций языка SAML будет истолковываться так же как синтаксис, семантика документа и правила обработки подтверждения той же версии. То есть набор спецификаций версии 1.0 описывает подтверждение версии 1.0 и т. д.

НЕ существует явной взаимосвязи между версией подтверждения и целевой областью имен XML, указанной для определения схемы версии этого подтверждения.

Применяются следующие правила обработки:

- подтверждающая сторона SAML не должна создавать никаких подтверждений с общим номером версии подтверждения *Major.Minor*, который не поддерживается ответственным органом;
- доверяющая сторона SAML не должна обрабатывать какие-либо подтверждения с номером основной версии подтверждения, который не поддерживается доверяющей стороной;
- доверяющая сторона SAML может обработать или может отбросить подтверждение, у которого номер дополнительной версии подтверждения выше номера дополнительной версии подтверждения, поддерживаемой доверяющей стороной. Однако все подтверждения, которые используют одну и ту же основную версию подтверждения, должны совместно использовать одни и те же общие правила обработки и семантику, и должны обеспечивать возможность их одинаковой обработки данным вариантом реализации. Например, если подтверждение V1.1 использует общий синтаксис подтверждения V1.0, вариант реализации может обрабатывать это подтверждение, как подтверждение V1.0 без нежелательного результата.

#### 8.3.1.3 Версия протокола SAML

Различные элементы запросов и ответов протокола SAML содержат атрибут для описания основной и дополнительной версии сообщения запроса или ответа, использующий строку вида *Major.Minor*. Каждая версия набора спецификаций языка SAML будет истолковываться так же как синтаксис, семантика документа и правила обработки подтверждения той же версии. То есть набор спецификаций версии 1.0 описывает подтверждение V1.0 и т. д.

Не существует явной взаимосвязи между версией протокола и целевой областью имен XML, указанной для определения схемы версии этого протокола.

Номера версий, используемые в элементах запросов и ответов протокола SAML, будут совпадать для любой конкретной пересмотренной версии набора спецификаций SAML.

**1) Версия запроса**

Применяются следующие правила обработки запросов:

- запрашивающая сторона SAML должна создавать запросы с наиболее высокой версией запроса, поддерживаемой и создателем запроса SAML, и отвечающей стороной SAML;
- если запрашивающая сторона SAML не знает возможностей отвечающей стороны SAML, то она должна предположить, что отвечающая сторона поддерживает запросы с наиболее высокой версией запроса, поддерживаемой запрашивающей стороной;
- запрашивающая сторона SAML не должна создавать сообщение-запрос с общим номером версии запроса *Major.Minor*, который совпадает с номером версии соответствующего ответа, если его не поддерживает запрашивающая сторона;
- отвечающая сторона SAML должна отбросить любые запросы с номером основной версии запроса, который не поддерживается отвечающей стороной.

Отвечающая сторона SAML может обработать или может отбросить любой запрос, у которого номер дополнительной версии запроса выше номера дополнительной версии подтверждения, запроса, которую эта сторона поддерживает. Однако все запросы, которые используют одну и ту же основную версию запроса, должны совместно использовать одни и те же общие правила обработки и семантику, и должны обеспечивать возможность их одинаковой обработки данным вариантом реализации. То есть если запрос V1.1 использует общий синтаксис запроса V1.0, отвечающая сторона может обрабатывать сообщение-запрос как запрос версии V1.0 без нежелательного результата.

**2) Версия ответа**

К ответам применяются следующие правила обработки:

- отвечающая сторона SAML не должна создавать сообщение-ответ с номером версии ответа большим, чем номер версии запроса соответствующего сообщения запроса;
- отвечающая сторона SAML не должна создавать сообщение-ответ с номером основной версии ответа, меньшим, чем номер версии запроса соответствующего сообщения запроса, за исключением сообщения об ошибке `urn:oasis:names:tc:SAML:2.0:status:RequestVersionTooHigh`;
- ошибочный ответ, полученный в результате несовместимых версий протокола SAML, должен привести к созданию значения кода состояния высшего уровня `<StatusCode>`, имеющего значение `urn:oasis:names:tc:SAML:2.0:status:VersionMismatch`, и может привести к передаче одного из следующих значений второго уровня:
  - `urn:oasis:names:tc:SAML:2.0:status:RequestVersionTooHigh`;
  - `urn:oasis:names:tc:SAML:2.0:status:RequestVersionTooLow`; или
  - `urn:oasis:names:tc:SAML:2.0:status:RequestVersionDeprecated`.

**3) Разрешенные комбинации версий**

Подтверждения определенной основной версии создаются только в сообщениях-ответах той же основной версии, что разрешено процедурой импортирования области имен подтверждений SAML в схему SAML протокола. Например, подтверждение V1.1 может быть создано в сообщении-ответе V1.0, и подтверждение V1.0 может быть создано в сообщении-ответе V1.1, если во время импортирования области имен указывается соответствующая схема подтверждения. Но подтверждение V1.0 не должно создаваться в сообщении-ответе V2.0, потому что это сообщения разных основных версий.

### 8.3.2 Версия области имен SAML

документация XML схемы, опубликованная в виде части набора спецификаций, содержит одну или несколько целевых областей имен, в которых размещаются определения типа, элемента и атрибута. Каждая область имен отличается от других областей и, вкратце, представляет собой структурные и синтаксические определения, которые образуют эту часть спецификации.

Ссылки на URI области имен, определенные набором спецификаций, будут, как правило, содержать информацию о версии в форме *Major.Minor* в каком-либо месте URI. Основная и дополнительная версия в URI должна соответствовать основной и дополнительной версии набора спецификаций, в котором эта область имен была впервые введена и определена. Эта информация обычно не используется процессором XML, который обрабатывает область имен непрозрачно, но она предназначена для осуществления взаимосвязи между набором спецификаций и областями имен, которые он определяет. Эта процедура выполняется также для идентификаторов, определенных в языке SAML на основе URI, которые перечислены в 8.7.

Согласно общему правилу разработчики могут ожидать, что области имен и связанная с ними схема определений, определенная в ходе крупного пересмотра набора спецификаций, останется действительной и неизменной при дополнительных пересмотрах спецификации. Могут быть введены новые области имен, и, при необходимости, могут быть заменены старые области имен, но этого не следует ожидать слишком часто. В таких ситуациях, следует ожидать, что более старые области имен и связанные с ними определения будут оставаться действительными до крупного пересмотра набора спецификаций.

Как правило, задачи поддержания стабильности области имен в ходе добавления или изменения содержания схемы являются противоречащими друг другу. Хотя определенные стратегические решения по планированию могут упростить такие изменения, очень сложно предсказать, как отреагируют более ранние варианты реализации на каждое данное изменение, что делает обеспечение совместимости вперед труднодостижимым. Тем не менее в интересах обеспечения стабильности области имен, зарезервировано право делать такие изменения при небольших пересмотрах. За исключением особых случаев (например, для корректировки больших разногласий или для исправления ошибок), следует ожидать, и при небольших пересмотрах будут вноситься изменения в схему совместимости вперед, позволяющие поверять достоверность новых сообщений с использованием старых схем.

Следует ожидать и следует быть готовым работать в условиях появления новых расширений и типов сообщений в соответствии с правилами обработки, установленными для этих типов. Небольшие пересмотры могут вводить новые типы, которые усиливают возможности расширения, описанные в настоящей Рекомендации. Более ранние варианты реализации должны мягко отбрасывать такие расширения, когда они их встречают в контексте, который определяет обязательную семантику. Например, новый тип вопроса, утверждения или условия.

## 8.4 Синтаксис и обработка подписи SAML и XML

Подтверждения SAML и сообщения запроса и ответа протокола SAML могут быть подписаны, что дает следующие выгоды. Подтверждение, подписанное доверяющей стороной, поддерживает целостность подтверждения, аутентификацию доверяющей стороны для доверяющей стороны SAML, и, если подпись основана на паре открытых ключей ответственного органа SAML, неотрекаемость источника сообщения. Сообщение запроса и ответа протокола SAML, подписанные создателем сообщения, поддерживает целостность сообщения, аутентификацию источника сообщения для адресата, и, если подпись основана на паре открытых ключей создателя сообщения, неотрекаемость источника сообщения.

Цифровая подпись не всегда требуется в языке SAML. Например, в некоторых условиях, подписи могут "наследоваться", например, когда неподписанное подтверждение получает защиту, обеспеченную подписью сообщения-ответа. "Унаследованные" подписи должны использоваться очень осторожно. Когда содержащийся объект (например, подтверждение) должен быть не временным объектом и меть продолжительное время существования. Причина в том, что для обеспечения достоверности подтверждения должен быть сохранен полный контекст, отображающий содержание XML с добавлением, возможно, лишнего заголовка. Другой пример – доверяющая сторона SAML или запрашивающая сторона SAML могли получить подтверждение или сообщение протокола от подтверждающей стороны SAML или непосредственно от отвечающей стороны SAML (без промежуточных этапов) по безопасному каналу, при этом доверяющая сторона или отвечающая сторона SAML были аутентифицированы для доверяющей стороны или отвечающей стороны SAML при помощи иных средств, отличных от цифровой подписи.

Существует множество различных методов для "прямой" аутентификации и установления безопасного канала между двумя сторонами. Это список включает в себя TLS, HMAC, механизмы на основе паролей и т. д. Кроме того, применимые требования по безопасности зависят от взаимодействующих приложений и природы передаваемого подтверждения или сообщения. Рекомендуется, чтобы, во всех других контекстах, для подтверждений и для сообщений запроса и ответа использовались цифровые подписи. В частности:

– подтверждение SAML, полученное доверяющей стороной SAML от элемента не являющегося подтверждающей стороной SAML, должно быть подписано подтверждающей стороной SAML;

– сообщение протокола SAML, прибывающее в пункт назначения от элемента не являющегося создателем сообщения, должно быть подписано отправителем;

– профили могут определять дополнительные механизмы подписи, например S/MIME или подписанные объекты Java, которые содержат документы SAML. Действуют предостережения о сохранении контекста и взаимодействия. Подписи XML являются основным механизмом подписи SAML, но в настоящей Рекомендации сделана попытка обеспечить совместимость с профилями, в которых могут требоваться другие механизмы;

– если профиль не определяет дополнительные механизмы подписи, должны использоваться любые цифровые подписи XML.

### 8.4.1 Подписание подтверждений

Все подтверждения SAML могут быть подписаны с использованием Правил подписи XML. Это отражено в схеме подтверждения, как описано в разделе 8.

### 8.4.2 Подписание запроса/ответа

Все сообщения запросов и ответов протокола SAML могут быть подписаны с использованием Правил подписи XML. Это отражено в схеме подтверждения, как описано в Приложении A.

### 8.4.3 Наследование подписи

Подтверждение SAML может быть вложено в другой элемент SAML, например, охватывающий элемент `<Assertion>` или запрос или ответ, который может быть подписан. Когда подтверждение SAML не содержит элемента `<ds:Signature>`, но он содержится в охватывающем его элементе SAML, который содержит элемент `<ds:Signature>`, и эта подпись относится к элементу `<Assertion>` и всем его дочерним элементам, тогда можно считать, что подтверждение наследует подпись от охватывающего его элемента. Итоговая интерпретация должна быть эквивалентная случаю, когда подписывается само подтверждение с тем же ключом и теми же опциями подписи.

Многие случаи использования языка SAML предусматривают, что данные SAML XML встраиваются в другие защищенные структуры данных, такие как подписанные сообщения SOAP, пакеты S/MIME и аутентифицированные соединения TLS. Профили SAML могут определять дополнительные правила интерпретации элементов SAML, как наследующих подписи или иную информацию аутентификации от окружающего их контекста, но такое наследование не должно предполагаться. Если оно не определено специально в профиле.

### 8.4.4 Профиль подписи XML

Документ "Правила подписи XML Консорциума W3C: 2002" содержит общий синтаксис XML для подписания данных, которому присуща гибкость и многовариантность. В настоящем разделе подробно рассмотрены ограничения этих возможностей, так чтобы процессоры SAML не были вынуждены работать с обширными возможностями обработки подписи XML. Такое применение определяет особый вариант использования атрибутов типа **xs:ID**, представленных в корневых элементах, к которым могут применяться подписи, в частности, атрибут `ID` в подтверждении `<Assertion>` и различные элементы запроса и ответа. Все эти атрибуты вместе в настоящем разделе называются атрибутами идентификатора.

Этот профиль применяется только для использования элементов `<ds:Signature>`, расположенных непосредственно внутри подтверждений, запросов и ответов SAML. Другие профили, в которых подписи могут быть в другом месте, но также относятся к содержанию SAML, могут определить другие подходы.

#### 8.4.4.1 Форматы и алгоритмы подписания

Подпись XML имеет три пути создания подписи, которые связывают ее с подписью документа: обертывание, конвертная и отсоединенная.

Подтверждения и протоколы SAML должны использовать конвертные подписи при подписании подтверждений и протокольных сообщений. Процессоры SAML должны поддерживать использование техники RSA для подписания и проверки достоверности для операций с открытым ключом в соответствии с алгоритмом, определенным в 6.4 документа http://www.w3.org/2000/09/xmldsig#rsa-sha1.

#### 8.4.4.2 Ссылки

Подтверждения и протокольные сообщения SAML должны представлять значение атрибута `ID` для подписываемого подтверждения корневого элемента или сообщения протокола. Корневой элемент сообщения подтверждения или протокола может быть, а может и не быть корневым элементом реального документа XML, содержащего подписанное подтверждение или протокольное сообщение (например, он может содержаться внутри формата SOAP).

Подписи должны содержать одну-единственную ссылку `<ds:Reference>`, содержащую ссылку на значение атрибута `ID` корневого элемента подписываемого подтверждения или передаваемого сообщения протокола этого же документа. Например, если значение атрибута `ID` = "foo", то атрибут `URI` в элементе `<ds:Reference>` должен быть = "#foo".

#### 8.4.4.3 Метод канонического назначения каналов

Варианты реализации SAML должны использовать метод Эксклюзивного канонического назначения каналов (Exclusive Canonicalization), с комментариями или без них, в обоих случаях – в элементе `<ds:CanonicalizationMethod>` объекта `<ds:SignedInfo>`, или в виде алгоритма `<ds:Transform>`. Использование метод Эксклюзивного канонического назначения каналов гарантирует, что подписи, созданные для сообщений SAML, встроенных в контекст XML, могут быть проверены независимо от этого контекста.

#### 8.4.4.4 Преобразования

Подписи в сообщениях SAML не должны содержать преобразований, кроме преобразования конвертной подписи (с идентификатором http://www.w3.org/2000/09/xmldsig#enveloped-signature) или преобразований эксклюзивного канонического назначения каналов (с идентификатором http://www.w3.org/2001/10/xml-exc-c14n# или http://www.w3.org/2001/10/xml-exc-c14n#WithComments).

Те, кто проверяет подписи, может отбросить подписи, которые содержат другие алгоритмы преобразования, как недействительные. Если те, кто проверяет подписи, это делают, они должны гарантировать, что никакая часть содержания сообщения SAML не будет выведена из-под действия подписи. Это может быть выполнено путем подписания внешнего соглашения о том, какие преобразования приемлемы, или путем ручного выполнения преобразований контекста и повторной проверки результата на соответствие тому же самому сообщению SAML.

#### 8.4.4.5 KeyInfo

Правила подписи W3C определяют использование элемента `<ds:KeyInfo>`. SAML не требует использования `<ds:KeyInfo>`, но и не налагает никаких ограничений на его использование. Следовательно, `<ds:KeyInfo>` может отсутствовать.

#### 8.4.4.6 Пример

Далее приведен пример подписанного ответа, содержащего подписанное подтверждение. Разрывы строк были добавлены для простоты чтения; подписи недействительны и их невозможно успешно проверить.

```
<Response
  IssueInstant="2003-04-17T00:46:02Z" Version="2.0"
  ID="_c7055387-af61-4fce-8b98-e2927324b306"
  xmlns="urn:oasis:names:tc:SAML:2.0:protocol"
  xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion">
 <saml:Issuer>https://www.opensaml.org/IDP"</saml:Issuer>
 <ds:Signature xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
    <ds:SignedInfo>
       <ds:CanonicalizationMethod
          Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#"/>
       <ds:SignatureMethod
          Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-sha1"/>
       <ds:Reference URI="#_c7055387-af61-4fce-8b98-e2927324b306">
          <ds:Transforms>
             <ds:Transform
                Algorithm="http://www.w3.org/2000/09/xmldsig#enveloped-
signature"/>
             <ds:Transform
                Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#">
                <InclusiveNamespaces PrefixList="#default saml ds xs xsi"
                   xmlns="http://www.w3.org/2001/10/xml-exc-c14n#"/>
             </ds:Transform>
          </ds:Transforms>
          <ds:DigestMethod
             Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
          <ds:DigestValue>TCDVSuG6grhyHbzhQFWFzGrxIPE=</ds:DigestValue>
       </ds:Reference>
    </ds:SignedInfo>
    <ds:SignatureValue>
       x/GyPbzmFEe85pGD3c1aXG4Vspb9V9jGCjwcRCKrtwPS6vdVNCcY5rHaFPYWkf+5
       EIYcPzx+pX1h43SmwviCqXRjRtMANWbHLhWAptaK1ywS7gFgsD01qjyen3CP+m3D
       w6vKhaqledl0BYyrIzb4KkHO4ahNyBVXbJwqv5pUaE4=
    </ds:SignatureValue>
    <ds:KeyInfo>
       <ds:X509Data>
          <ds:X509Certificate>
```

```
         MIICyjCCAjOgAwIBAgICAnUwDQYJKoZIhvcNAQEEBQAwgakxCzAJBgNVBAYTAlVT

         MRIwEAYDVQQIEwlXaXNjb25zaW4xEDAOBgNVBAcTB01hZGlzb24xIDAeBgNVBAoT

         F1VuaXZlcnNpdHkgb2YgV2lzY29uc2luMSswKQYDVQQLEyJEaXZpc2lvbiBvZiBJ

         bmZvcm1hdGlvbiBUZWNobm9sb2d5MSUwIwYDVQQDExxIRVBLSSBTZXJ2ZXIgQ0Eg

         LS0gMjAwMjA3MDFBMB4XDTAyMDcyNjA3Mjc1MVoXDTA2MDkwNDA3Mjc1MVowgYsx

         CzAJBgNVBAYTAlVTMREwDwYDVQQIEwhNaWNoaWdhbjESMBAGA1UEBxMJQW5uIEFy

         Ym9yMQ4wDAYDVQQKEwVVQ0FJRDEcMBoGA1UEAxMTc2hpYjEuaW50ZXJuZXQyLmVk

         dTEnMCUGCSqGSIb3DQEJARYYcm9vdEBzaGliMS5pbnRlcm5ldDIuZWR1MIGfMA0G

         CSqGSIb3DQEBAQUAA4GNADCBiQKBgQDZSAb2sxvhAXnXVIVTx8vuRay+x50z7GJj

         IHRYQgIv6IqaGG04eTcyVMhoekE0b45QgvBIaOAPSZBl13R6+KYiE7x4XAWIrCP+

         c2MZVeXeTgV3Yz+USLg2Y1on+Jh4HxwkPFmZBctyXiUr6DxF8rvoP9W7O27rhRjE

         pmqOIfGTWQIDAQABox0wGzAMBgNVHRMBAf8EAjAAMAsGA1UdDwQEAwIFoDANBgkq

         hkiG9w0BAQQFAAOBgQBfDqEW+OI3jqBQHIBzhujN/PizdN7s/z4D5d3pptWDJf2n

         qgi7lFV6MDkhmTvTqBtjmNk3No7v/dnP6Hr7wHxvCCRwubnmIfZ6QZAv2FU78pLX

         8I3bsbmRAUg4UP9hH6ABVq4KQKMknxu1xQxLhpR1ylGPdiowMNTrEG8cCx3w/w==

         </ds:X509Certificate>
      </ds:X509Data>
   </ds:KeyInfo>
 </ds:Signature>
 <Status>
   <StatusCode Value="urn:oasis:names:tc:SAML:2.0:status:Success"/>
 </Status>
 <Assertion ID="_a75adf55-01d7-40cc-929f-dbd8372ebdfc"
   IssueInstant="2003-04-17T00:46:02Z" Version="2.0"
   xmlns="urn:oasis:names:tc:SAML:2.0:assertion">
   <Issuer>https://www.opensaml.org/IDP</Issuer>
   <ds:Signature xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
      <ds:SignedInfo>
         <ds:CanonicalizationMethod
            Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#"/>
         <ds:SignatureMethod
            Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-sha1"/>
         <ds:Reference URI="#_a75adf55-01d7-40cc-929f-dbd8372ebdfc">
            <ds:Transforms>
               <ds:Transform
                  Algorithm="http://www.w3.org/2000/09/xmldsig#enveloped-
signature"/>
               <ds:Transform
                  Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#">
                  <InclusiveNamespaces
                     PrefixList="#default saml ds xs xsi"
                     xmlns="http://www.w3.org/2001/10/xml-exc-c14n#"/>
               </ds:Transform>
            </ds:Transforms>
            <ds:DigestMethod
               Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
            <ds:DigestValue>Kclet6XcaOgOWXM4gty6/UNdviI=</ds:DigestValue>
         </ds:Reference>
      </ds:SignedInfo>
      <ds:SignatureValue>
         hq4zk+ZknjggCQgZm7ea8fI79gJEsRy3E8LHDpYXWQIgZpkJN9CMLG8ENR4Nrw+n
         7iyzixBvKXX8P53BTCT4VghPBWhFYSt9tHWu/AtJfOTh6qaAsNdeCyG86jmtp3TD
         MwuL/cBUj2OtBZOQMFn7jQ9YB7klIz3RqVL+wNmeWI4=
      </ds:SignatureValue>
```

```
      <ds:KeyInfo>
         <ds:X509Data>
            <ds:X509Certificate>
         MIICyjCCAjOgAwIBAgICAnUwDQYJKoZIhvcNAQEEBQAwgakxCzAJBgNVBAYTAlVT

         MRIwEAYDVQQIEwlXaXNjb25zaW4xEDAOBgNVBAcTB01hZGlzb24xIDAeBgNVBAoT

         F1VuaXZlcnNpdHkgb2YgV2lzY29uc2luMSswKQYDVQQLEyJEaXZpc2lvbiBvZiBJ

         bmZvcm1hdGlvbiBUZWNobm9sb2d5MSUwIwYDVQQDExxIRVBLSSBTZXJ2ZXIgQ0Eg

         LS0gMjAwMjA3MDFBMB4XDTAyMDcyNjA3Mjc1MVoXDTA2MDkwNDA3Mjc1MVowgYsx

         CzAJBgNVBAYTAlVTMREwDwYDVQQIEwhNaWNoaWdhbjESMBAGA1UEBxMJQW5uIEFy

         Ym9yMQ4wDAYDVQQKEwVVQ0FJRDEcMBoGA1UEAxMTc2hpYjEuaW50ZXJuZXQyLmVk

         dTEnMCUGCSqGSIb3DQEJARYYcm9vdEBzaGliMS5pbnRlcm5ldDIuZWR1MIGfMA0G

         CSqGSIb3DQEBAQUAA4GNADCBiQKBgQDZSAb2sxvhAXnXVIVTx8vuRay+x50z7GJj

         IHRYQgIv6IqaGG04eTcyVMhoekE0b45QgvBIaOAPSZBl13R6+KYiE7x4XAWIrCP+

         c2MZVeXeTgV3Yz+USLg2Y1on+Jh4HxwkPFmZBctyXiUr6DxF8rvoP9W7O27rhRjE

         pmqOIfGTWQIDAQABox0wGzAMBgNVHRMBAf8EAjAAMAsGA1UdDwQEAwIFoDANBgkq

         hkiG9w0BAQQFAAOBgQBfDqEW+OI3jqBQHIBzhujN/PizdN7s/z4D5d3pptWDJf2n

         qgi7lFV6MDkhmTvTqBtjmNk3No7v/dnP6Hr7wHxvCCRwubnmIfZ6QZAv2FU78pLX

         8I3bsbmRAUg4UP9hH6ABVq4KQKMknxu1xQxLhpR1ylGPdiowMNTrEG8cCx3w/w==

            </ds:X509Certificate>
         </ds:X509Data>
      </ds:KeyInfo>
   </ds:Signature>
   <Subject>
      <NameID
         Format="urn:oasis:names:tc:SAML:1.1:nameid-format:emailAddress">
         scott@example.org
      </NameID>
      <SubjectConfirmation
         Method="urn:oasis:names:tc:SAML:2.0:cm:bearer"/>
   </Subject>
   <Conditions NotBefore="2003-04-17T00:46:02Z"
         NotOnOrAfter="2003-04-17T00:51:02Z">
      <AudienceRestriction>
         <Audience>http://www.opensaml.org/SP</Audience>
      </AudienceRestriction>
   </Conditions>
   <AuthnStatement AuthnInstant="2003-04-17T00:46:00Z">
      <AuthnContext>
         <AuthnContextClassRef>
            urn:oasis:names:tc:SAML:2.0:ac:classes:Password
         </AuthnContextClassRef>
      </AuthnContext>
   </AuthnStatement>
 </Assertion>
</Response>
```

## 8.5 Синтаксис и обработка шифрования SAML и XML

Шифрование используется как средство обеспечения конфиденциальности. Наиболее широко распространенными причинами требований к конфиденциальности являются защита частных секретов человека и защита секретов организации с целью получения преимуществ в конкурентной борьбе или по аналогичным причинам. Конфиденциальность может также требоваться для обеспечения эффективности некоторых других механизмов безопасности. Например, секретный пароль или ключ могут быть зашифрованы.

Имеется несколько способов использовать шифрование для защиты конфиденциальности всего подтверждения SAML целиком или отдельных его частей:

- конфиденциальность передачи может быть обеспечена механизмами, ассоциированными с определенной связью или профилем. Например, связь SOAP поддерживает использование механизмов безопасности TLS (см. Документ IETF RFC 2246) или сообщения SOAP для обеспечения конфиденциальности;
- секрет `<SubjectConfirmation>` может быть защищен при помощи использования элемента `<ds:KeyInfo>` внутри `<SubjectConfirmationData>`, что позволяет шифровать ключи или иные секреты;
- элемент `<Assertion>` может быть зашифрован целиком, как описано в 8.1.3.4;
- элемент `<BaseID>` или `<NameID>` может быть зашифрован, как описано в 8.1.2.4;
- элемент `<Attribute>` может быть зашифрован, как описано в 8.1.7.3.2.

### 8.5.1 Общие соображения

Шифрование элементов `<Assertion>`, `<BaseID>`, `<NameID>` и `<Attribute>` выполняется с использованием XML шифрования. Шифрованные данные и, возможно, один или несколько зашифрованных ключей должны заменить информацию, передаваемую в виде обычного текста внутри элемента XML. Должен использоваться атрибут типа элемента `<EncryptedData>` и, если он представлен, он должен иметь значение http://www.w3.org/2001/04/xmlenc#Element.

ПРИМЕЧАНИЕ (информативное). – PE30 (см. OASIS PE:2006) предлагает заменить слова "один или несколько" во второй строчке словами "ноль или более".

Для выполнения шифрования может использоваться любой из алгоритмов, определенных с использованием XML шифрования. Схема SAML определяется так, что введение в нее зашифрованных данных приводит к получению действительного значения.

### 8.5.2 Объединение подписей и шифрования

Можно скомбинировать использование шифрования XML и подписи XML. Когда подтверждение должно быть подписано и зашифровано, применяются следующие правила. Доверяющая сторона должна выполнить проверку достоверности подписи и ее дешифрование в порядке, обратном тому, который использовался при выполнении подписания и шифрования.

- Когда шифруется подписанный элемент `<Assertion>`, подпись должна быть сначала рассчитана и помещена внутрь элемента `<Assertion>` до его шифрования.
- Когда шифруется элемент `<BaseID>`, `<NameID>` или `<Attribute>`, сначала должно быть выполнено шифрование, а затем рассчитана подпись подтверждения или сообщения, содержащего зашифрованный элемент.

## 8.6 Возможность расширения SAML

SAML поддерживает разные способы расширения, включая расширение схем подтверждения и протокола. Информация о том, как определить новые профили, которые можно будет комбинировать с расширениями для формирования новой среды SAML для новых пользователей, приведена в разделе Профили настоящей Рекомендации.

### 8.6.1 Расширение схемы

Элементы в схеме SAML блокированы от замен, что означает, что ни один из элементов SAML не может считаться главным элементом группы замены. Однако типы SAML окончательно еще не определены, поэтому все типы SAML могут быть расширены и ограничены. С практической точки зрения это означает, что расширения обычно определяются только как типы, а не как элементы, и они в язык SAML включены экземпляры при помощи атрибута `xsi:type`.

В последующих подразделах рассматриваются только те элементы и типы, которые быть специально разработаны для поддержания возможности расширений.

#### 8.6.1.1 Расширение схемы подтверждения

Схема подтверждения SAML разработана для обеспечения отдельной обработки пакетов подтверждений и утверждений, которые он содержит, если для любой из частей используется механизм расширения.

Следующие элементы специально предназначены для использования в качестве точек расширения в схеме расширения; их типы переданы в элемент `abstract`, и, следовательно, могут использоваться только как основа для производного типа:

- `<BaseID>` и **BaseIDAbstractType**;
- `<Condition>` и **ConditionAbstractType**;

- `<Statement>` и **StatementAbstractType**.

Далее приведены конструкции, которые непосредственно могут использоваться как часть SAML, они являются особенно интересными целями для расширения:

- `<AuthnStatement>` и **AuthnStatementType**;
- `<AttributeStatement>` и **AttributeStatementType**;
- `<AuthzDecisionStatement>` и **AuthzDecisionStatementType**;
- `<AudienceRestriction>` и **AudienceRestrictionType**;
- `<ProxyRestriction>` и **ProxyRestrictionType**;
- `<OneTimeUse>` и **OneTimeUseType**.

#### 8.6.1.2 Расширение схемы протокола

Следующие элементы схемы протокола SAML предназначены специально для использования точек расширения в схеме расширения; их типы установлены абстрактными, и, таким образом, могут использоваться только как основа для производного типа:

- `<Request>` и **RequestAbstractType**;
- `<SubjectQuery>` и **SubjectQueryAbstractType**.

Следующие конструкции, которые могут быть использованы непосредственно как часть SAML, наиболее интересная цель расширения:

- `<AuthnQuery>` и **AuthnQueryType**;
- `<AuthzDecisionQuery>` и **AuthzDecisionQueryType**;
- `<AttributeQuery>` и **AttributeQueryType**;
- **StatusResponseType**.

### 8.6.2 Точки расширения подстановочных знаков схемы

В некоторых местоположениях схемы SAML применяют конструкции подстановочных знаков для того, чтобы иметь возможность использовать элементы и атрибуты из произвольных областей имен, такая конструкция работает, как строенная точка расширения без необходимости использовать схему расширения.

#### 8.6.2.1 Точки расширения подтверждения

Приведенные далее конструкции в схеме подтверждения позволяют иметь внутри себя конструкции из произвольных областей имен:

- `<SubjectConfirmationData>`: Использует тип **xs:anyType**, который позволяет иметь субэлементы и атрибуты.
- `<AuthnContextDecl>`: Использует тип **xs:anyType**, который позволяет существование любых субэлементов и атрибутов.
- `<AttributeValue>`: Использует тип **xs:anyType**, который позволяет существование любых субэлементов и атрибутов.
- `<Advice>` и **AdviceType**: В дополнение к элементами, созданным в языке SAML, позволяет работать с элементами из других областей имен, при помощи схемы обработки достоверности.

Следующая конструкция в подтверждении схемы позволяет работать с глобальными атрибутами:

- `<Attribute>` и тип атрибута **Type**.

#### 8.6.2.2 Точки расширения протокола

Приведенные далее конструкции в схеме протокола позволяют иметь внутри себя конструкции из произвольных областей имен:

- `<Extensions>` и тип **ExtensionsType**: Позволяет вводить элементы из других областей имен с мягкой обработкой подтверждения достоверности схемы.
- `<StatusDetail>` и тип **StatusDetailType**: Позволяет вводить элементы из других областей имен с мягкой обработкой подтверждения достоверности схемы.
- `<ArtifactResponse>` и тип **ArtifactResponseType**: Позволяет вводить элементы из других областей имен с мягкой обработкой подтверждения достоверности схемы. (Однако он специально предназначен для передачи элементов сообщений запросов или ответов SAML.)

### 8.6.3 Расширение идентификатора

В языке SAML идентификаторы на основе URI используются для различных целей, например в качестве форматов кодов состояния или идентификатора имени, и в языке определены некоторые идентификаторы, которые могут использоваться для этих целей; большая их часть перечислена в 8.7. Однако для этих целей всегда можно определить дополнительные идентификаторы на основе URI. Рекомендуется, чтобы эти дополнительные идентификаторы были бы определены в формате используемого профиля. Ни при каких обстоятельствах значение данного URI, используемого в качестве такого идентификатора не должно существенно изменяться, или не должно использоваться для обозначения двух различных вещей.

## 8.7 Идентификаторы, определенные в SAML

В последующих подразделах определяются идентификаторы на основе URI для действий доступа к общим ресурсам, зависящие от форматов идентификатора имени и форматов называний атрибутов.

Везде, где это возможно, для определения протокола используется существующий URN. Для протоколов IETF используется URN из последнего документа RFC, который определяет этот протокол. Ссылки на URN, созданные специально для языка SAML, имеют одну из следующих основ в соответствии с версией набора спецификаций, в котором они были впервые введены:

```
urn:oasis:names:tc:SAML:1.0:
urn:oasis:names:tc:SAML:1.1:
urn:oasis:names:tc:SAML:2.0:
```

Настоящая Рекомендация вводит последнюю основу.

### 8.7.1 Идентификаторы области имен Action

Следующие идентификаторы могут использоваться в атрибуте области имен элемента `<Action>` для указания на общие наборы действий, которые можно выполнить над данными ресурсами.

#### 8.7.1.1 Read/Write/Execute/Delete/Control

**URI**: `urn:oasis:names:tc:SAML:1.0:action:rwedc`

Определяемые действия: `Read Write Execute Delete Control`

Эти действия интерпретируются следующим образом:

`Read`: Объект может читать ресурс.

`Write`: Объект может модифицировать ресурс.

`Execute`: Объект может выполнить ресурс.

`Delete`: Объект может удалить ресурс.

`Control`: Объект может определить политику регулирования доступа к ресурсу.

#### 8.7.1.2 Read/Write/Execute/Delete/Control с отрицательным значением

**URI**: `urn:oasis:names:tc:SAML:1.0:action:rwedc-negation`

Определяемые действия: `Read Write Execute Delete Control ~Read ~Write ~Execute ~Delete ~Control`

Действия, определенные в 8.7.1.1 интерпретируются, как описано здесь. Действия, которым предшествует префикс тильда (~) представляют собой отрицательные разрешения и используются для того, чтобы четко определить, что это разрешение отменено. Таким образом, объект, описываемый, как имеющий разрешение на выполнение действия ~Read, совершенно четко не имеет разрешение на чтение.

Ответственный орган SAML не должен разрешать выполнение и самого действия и его отрицательной формы.

#### 8.7.1.3 Get/Head/Put/Post

**URI**: `urn:oasis:names:tc:SAML:1.0:action:ghpp`

Определяемые действия: `GET HEAD PUT POST`

Эти действия связаны с соответствующими операциями HTTP. Например, объект, которому разрешено выполнение действия `GET` над ресурсом, имеет право его получить.

Действия `GET` и `HEAD` некоторым образом соответствуют обычному разрешению на чтение, а действия `PUT` и `POST` – разрешению на модификацию. Однако это соответствие не совсем точное, поскольку операция HTTP GET может привести к модификации данных, а операция POST может привести к модификации ресурса, отличного от того, который указан в запросе. По этой причине предусмотрен спецификатор ссылок на действия URI.

#### 8.7.1.4 Разрешения для файлов UNIX

**URI**: `urn:oasis:names:tc:SAML:1.0:action:unix`

Определяемыми действиями является набор разрешений на доступ к файлам UNIX, выраженный в цифровой (восьмеричной) нотации.

Строка действий – это четырехзначный цифровой код:

*extended user group world*

Где разрешение доступа *extended* имеет следующее значение:

+2, если установлен sgid;

+4, если установлен suid;

Разрешение доступа *user group* и *world* имеют следующие значения:

+1, если дано разрешение на выполнение;

+2, если дано разрешение на модификацию;

+4, если дано разрешение на чтение.

Например, 0754 обозначает следующее разрешение на доступ к файлам UNIX: пользователь может читать, модифицировать и выполнять; группа может читать, модифицировать и выполнять; все остальные могут читать.

### 8.7.2 Идентификаторы формата имени атрибута

Следующие идентификаторы могут использоваться в атрибуте `NameFormat`, определенном в сложном типе **AttributeType,** для обозначения классификации имени атрибута для целей интерпретации имени.

#### 8.7.2.1 Не определен

**URI**: `urn:oasis:names:tc:SAML:2.0:attrname-format:unspecified`

Интерпретация имени атрибута зависит от конкретного варианта реализации.

#### 8.7.2.2 Ссылка на URI

**URI**: `urn:oasis:names:tc:SAML:2.0:attrname-format:uri`

Имя атрибута соответствует конвенции по ссылкам на URI, например как используемые в идентификаторах атрибута XACML. Интерпретация содержания URI или схемы имен зависит от приложения. Профили атрибутов, которые должны использовать этот идентификатор, приведены в разделе 11.

#### 8.7.2.3 Основа

**URI**: `urn:oasis:names:tc:SAML:2.0:attrname-format:basic`

Этот класс строк, приемлемых в качестве имени атрибута, должен быть взят из множества значений, принадлежащих элементарному типу **xs:Name**, как определено в 3.3.6 документ "Типы XML данных Консорциума W3C. Профили атрибутов, которые должны использовать этот идентификатор, приведены в разделе 13.

### 8.7.3 Идентификатор имени формата идентификатора имени

Следующие идентификаторы могут использоваться в атрибуте `Format` элементов `<NameID>` `<NameIDPolicy>` или `<Issuer>` (см. 8.1.2) для обозначения общих форматов для содержания элемента и связанных с ними правил обработки, если таковые имеются.

ПРИМЕЧАНИЕ. – Некоторые идентификаторы, которые встретили возражения в языке SAML V1.1, были удалены при создании второй версии языка SAML V2.0.

#### 8.7.3.1 Не определен

**URI**: `urn:oasis:names:tc:SAML:1.1:nameid-format:unspecified`

Интерпретация содержания элемента зависит от конкретного варианта реализации.

#### 8.7.3.2 Адрес электронной почты

**URI**: `urn:oasis:names:tc:SAML:1.1:nameid-format:emailAddress`

Указывает, что содержание элемента имеет форму адреса электронной почты, в частности "addr-spec", определенную в IETF RFC 2822, 3.4.1. Форма addr-spec имеет вид local-part@domain. Отметим, что форма addr-spec не содержит перед собой фразы (типа обычного имени), не содержит комментариев (текст в скобках) после себя, и не окружена знаками "<" и ">".

#### 8.7.3.3 Имя объекта X.509

**URI**: urn:oasis:names:tc:SAML:1.1:nameid-format:X509SubjectName

Указывает, что содержание элемента имеет форму, определенную для содержания элемента <ds:X509SubjectName> в Правилах подписи W3. Разработчики должны помнить, что Правила XML подписи определяют правила кодирования для имен объектов X.509 и отличаются от правил, описанных в IETF RFC 2253.

#### 8.7.3.4 Имя, определенное в домене Windows

**URI**: urn:oasis:names:tc:SAML:1.1:nameid-format:WindowsDomainQualifiedName

Указывает, что содержанием элемента является имя, определенное в домене Windows. Имя пользователя, определенное в домене Windows, это – строка вида "Имя Домена\Имя Пользователя". Имя Домена и разделитель "\" могут быть пропущены.

#### 8.7.3.5 Имя клиента Kerberos

**URI**: urn:oasis:names:tc:SAML:2.0:nameid-format:kerberos

Указывает, что содержание элемента имеет форму имени клиента Kerberos в формате name[/instance]@REALM. Синтаксис, формат и характеристики, допустимые для этого имени, экземпляра и рабочей области описаны в IETF RFC 1510.

#### 8.7.3.6 Идентификатор элемента

**URI**: urn:oasis:names:tc:SAML:2.0:nameid-format:entity

Указывает, что содержание элемента – это идентификатор элемента, который предоставляет услуги на базе SAML (например, ответственный орган SAML, запрашивающая сторона или отвечающая сторона) или участник профиля SAML (например, провайдер услуг, поддерживающий профиль браузера). Такой идентификатор может использоваться в элементе <Issuer> для идентификации создателя запроса, ответа или подтверждения SAML, или в элементе <NameID> для формирования подтверждений об элементах системы, которые могут создавать запросы, ответы или подтверждения SAML. Он также может использоваться в других элементах и атрибутах, чьей целью является идентификация элемента системы в ходе различных протокольных передач.

Синтаксис такого идентификатора – это URI длиной не более 1024 символов. Рекомендуется, чтобы элемент системы использовал URL, содержащий собственное доменное имя для идентификации самого себя.

Атрибуты NameQualifier, SPNameQualifier и SPProvidedID должны быть пропущены.

#### 8.7.3.7 Постоянный идентификатор

**URI**: urn:oasis:names:tc:SAML:2.0:nameid-format:persistent

Указывает, что содержанием элемента является постоянный скрытый идентификатор клиента, который зависит от провайдера идентификации и провайдера услуг или объединения провайдеров услуг. Постоянные идентификаторы имен, создаваемые провайдерами идентификации, должны формироваться с использованием псевдослучайных величин, которые не имеют видимого соответствия с действительным идентификатором объекта (например, именем пользователя). Целью является создание парного псевдонима, не доступного широкой публике, который дал бы возможность предотвратить раскрытие имени или действий объекта. Значения постоянного идентификатора имени должны иметь длину не более 256 символов.

Атрибут элемента NameQualifier, если представлен, должен содержать уникальный идентификатор провайдера идентификации, который создал этот идентификатор (см. 8.7.3.6). Он может быть пропущен, если это значение может быть получено из контекста сообщения, содержащего такой элемент, как создатель сообщения протокола или подтверждения, содержащего идентификатор в самом объекте. Другой элемент системы может позже создать свое собственное сообщение протокола или подтверждение, содержащее идентификатор; атрибут NameQualifier в этом случае не меняется, но должен продолжать идентифицировать элемент, который изначально создал этот идентификатор (и в таком случае он не должен быть пропущен).

Атрибут элемента SPNameQualifier, если представлен, должен содержать уникальный идентификатор провайдера услуг или объединения провайдеров, для которых был создан этот идентификатор (см. 8.7.3.6). Он может быть пропущен, если этот элемент содержится в сообщении, предназначенном только для непосредственного использования провайдером услуг, и его значением будет уникальный идентификатор этого провайдера услуг.

Атрибут элемента SPProvidedID должен содержать дополнительный идентификатор клиента, последний (по времени), установленный провайдером услуг или их объединением, если таковой имеется (см. 8.2.6). Если такой идентификатор не установлен, то этот атрибут должен быть пропущен.

Постоянные идентификаторы выполняют роли механизмов защиты секретности; и, раз так, они не должны встречаться в незашифрованном тексте при переписке с провайдерами, с которыми не был создан идентификатор для совместного использования. Кроме того, они не должны встречаться в файлах регистрации или подобных местах без соответствующих мер контроля и защиты. В тех вариантах реализации, когда такие требования не предъявляются, могут использовать в ходе обмена сообщениями SAML другие типы идентификаторов, но они не должен перегружать этот формат, вводя в него постоянные, но не скрытые значения.

В то время как постоянные идентификаторы обычно используются для отображения линий взаимных связей между парой провайдеров для данного аккаунта, провайдер услуг не обязан распознавать или использовать для создания таких линий долгосрочную природу постоянного идентификатора. Такие "односторонние" отношения не являются явно иными и не влияют на действия провайдера идентификации или какие-либо правила обработки постоянных идентификаторов в протоколах, определенных в настоящей Рекомендации.

Атрибуты `NameQualifier` и `SPNameQualifier` непосредственно указывают создание, но не использование. Если постоянный идентификатор создается определенным провайдером идентификации, то в это время значение атрибута `NameQualifier` должно быть постоянно установленным. Если провайдер услуг, который получает такой идентификатор, играет роль провайдера идентификации и создает свое собственное подтверждение, содержащее этот идентификатор, то значение атрибута `NameQualifier` не меняется (и, конечно, не будет пропущено). Он может принять решение о создании собственного постоянного идентификатора для представления клиента и установления связи между двумя этими значениями. Это решение зависит от варианта реализации.

#### 8.7.3.8 Временный идентификатор

**URI**: `urn:oasis:names:tc:SAML:2.0:nameid-format:transient`

Указывает, что содержанием элемента является идентификатор с временной семантикой и должен обрабатываться доверяющей стороной как скрытое и временное значение. Значения временного идентификатора должны создаваться в соответствии с правилами для идентификаторов языка SAML (см. 7.4) и не должны превышать длину в 256 символов.

Атрибуты `NameQualifier` и `SPNameQualifier` могут использоваться для указания того, что идентификатор представляет собой временный парный идентификатор. В таком случае они могут быть пропущены в соответствии с правилами, определенными в 8.7.3.7.

### 8.7.4 Идентификаторы соглашения

Следующие идентификаторы могут использоваться в атрибуте `Consent`, определенном в сложных типах **RequestAbstractType** и **StatusResponseType** для сообщения о том, дал ли согласие клиент для этого сообщения и на каких условиях.

#### 8.7.4.1 Неопределенный

**URI**: `urn:oasis:names:tc:SAML:2.0:consent:unspecified`

Никаких заявлений о согласии клиента не было.

#### 8.7.4.2 Полученный

**URI**: `urn:oasis:names:tc:SAML:2.0:consent:obtained`

Указывает, что создателем сообщения получено согласие клиента.

#### 8.7.4.3 Предварительный

**URI**: `urn:oasis:names:tc:SAML:2.0:consent:prior`

Указывает, что создателем сообщения получено согласие клиента в некоторый момент, предшествующий действию, результатом которого стало это сообщение.

#### 8.7.4.4 Неявный

**URI**: `urn:oasis:names:tc:SAML:2.0:consent:current-implicit`

Указывает, что создателем сообщения получено неявное согласие клиента во время действия, результатом которого стало это сообщение, как часть более широкого соглашения. Неявное согласие, как правило, является более близким, чем предварительное согласие, к действию и представлению, например, части сеанса действий.

#### 8.7.4.5 Явный

**URI**: `urn:oasis:names:tc:SAML:2.0:consent:current-explicit`

Указывает, что создателем сообщения получено явное согласие клиента во время действия, результатом которого стало это сообщение.

#### 8.7.4.6 Недоступный

**URI**: `urn:oasis:names:tc:SAML:2.0:consent:unavailable`

Указывает, что создателем сообщения не получено согласие.

#### 8.7.4.7 Неприменимый

**URI**: `urn:oasis:names:tc:SAML:2.0:consent:inapplicable`

Указывает, что создатель сообщения не считает, что ему требуется получать согласие или сообщать о нем.

## 9 Метаданные SAML

Профили языка SAML требуют соглашения между элементами системы относительно идентификаторов, поддержки связи и оконечных точек, сертификатов и ключей и т. д. В настоящем разделе определяется расширяемый формат метаданных для элементов системы языка SAML, организованный по ролям, которые отражают профили SAML. Такие роли включают в себя роли SSO провайдера идентификации, SSO провайдера услуг, Объединение, Ответственный орган атрибута, Запрашивающая сторона атрибута и Точка принятия стратегического решения.

### 9.1 Метаданные

Метаданные SAML организованы вокруг расширяемого множества ролей, представляющих собой общие объединения протоколов SAML и профилей, поддерживаемых элементами системы. Каждая роль описывается элементом, являющимся производным от расширяемого базового типа `RoleDescriptor`. Такие описатели, в свою очередь, собраны в элементе контейнере `<EntityDescriptor>`, элементарной единице Метаданных SAML. Элемент может представлять объединение других элементов, например, объединение провайдеров услуг. Для этой цели предусмотрен элемент `<AffiliationDescriptor>`.

Такие описатели, в свою очередь, могут быть объединены во вложенные группы с использованием элемента `<EntitiesDescriptor>`.

Может поддерживаться множество разнообразных механизмов безопасности для установления достоверности метаданных, в частности, механизмы, имеющие возможность подписывать отдельно большую часть элементов, определенных в настоящей Рекомендации.

Когда элементы, имеющие взаимосвязь типа родитель-дитя, содержат общие атрибуты, такие как кэширование или истечение срока действия информация, преимущество имеет элемент "родитель".

ПРИМЕЧАНИЕ. – В общем смысле метаданные SAML не должны восприниматься как авторитетное утверждение о способностях и возможностях данного элемента системы. То есть несмотря на то что они должны быть очными, они не должны быть исчерпывающими. Пропуск определенной опции не означает, что она поддерживается или не поддерживается, это означает только, что об этом ничего не сказано. Например, Ответственный орган атрибута SAML может поддерживать любое количество атрибутов, не названных в `<AttributeAuthorityDescriptor>`. Пропуск атрибутов может быть обусловлен требованиями секретности или любыми другими соображениями. Напротив, указание поддержки для данного атрибута не означает, что данная запрашивающая сторона сможет или будет принимать его.

#### 9.1.1 Области имен

Метаданные SAML используют следующую область имен:

```
urn:oasis:names:tc:SAML:2.0:metadata
```

В настоящей Рекомендации используется префикс области имен `md:` для обозначения области имен, показанной выше.

Приведенный далее фрагмент схемы иллюстрирует использование области имен в документах метаданных языка SAML:

```
<schema
    targetNamespace="urn:oasis:names:tc:SAML:2.0:metadata"
    xmlns:md="urn:oasis:names:tc:SAML:2.0:metadata"
    xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
    xmlns:xenc="http://www.w3.org/2001/04/xmlenc#"
    xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion"
    xmlns="http://www.w3.org/2001/XMLSchema"
    elementFormDefault="unqualified"
    attributeFormDefault="unqualified"
    blockDefault="substitution"
    version="2.0">
    <import namespace="http://www.w3.org/2000/09/xmldsig#"
        schemaLocation="http://www.w3.org/TR/2002/REC-xmldsig-core-
20020212/xmldsig-core-schema.xsd"/>
    <import namespace="http://www.w3.org/2001/04/xmlenc#"
        schemaLocation="http://www.w3.org/TR/2002/REC-xmlenc-core-
20021210/xenc-schema.xsd"/>
    <import namespace="urn:oasis:names:tc:SAML:2.0:assertion"
```

```
        schemaLocation="saml-schema-assertion-2.0.xsd"/>
    <import namespace="http://www.w3.org/XML/1998/namespace"
        schemaLocation="http://www.w3.org/2001/xml.xsd"/>
    <annotation>
        <documentation>
            Document identifier: saml-schema-metadata-2.0
            Location: http://docs.oasis-open.org/security/saml/v2.0/
            Revision history:
              V2.0 (March, 2005):
                Schema for Metadata SAML, first published in SAML 2.0.
        </documentation>
    </annotation>
…
</schema>
```

### 9.1.2 Общие типы

В настоящем разделе определяется несколько типов метаданных, которые должны использоваться при определении элементов и атрибутов.

#### 9.1.2.1 Простой тип entityIDType

Простой тип **entityIDType** ограничивает максимальную длину данных типа **anyURI** в схеме XML величиной 1024 символами. Тип **entityIDType** используется как уникальный идентификатор для элементов SAML (см. также 8.7.3.6). Идентификатор этого типа должен быть уникальным для всех элементов, с которыми он взаимодействует в пределах данной системы. Использование URI и соблюдение правила, согласно которому один-единственный URI не должен обозначать различные элементы, удовлетворяет этому требованию.

Приведенный далее фрагмент схемы определяет простой тип **IDType**:

```
<simpleType name="entityIDType">
 <restriction base="anyURI">
    <maxLength value="1024"/>
 </restriction>
</simpleType>
```

#### 9.1.2.2 Сложный тип EndpointType

Сложный тип **EndpointType** описывает оконечную точку связи протокола SAML, от которой элемент SAML может передавать протокольные сообщения. С этим типом связаны различные элементы метаданных, определяемые протоколом или профилем. Он состоит из следующих атрибутов:

– `Binding` [Требуемый]

  Требуемый атрибут, который определяет связь SAML, поддерживаемую оконечной точкой. Каждой связи назначается URI, который ее идентифицирует.

– `Location` [Требуемый]

  Требуемый атрибут URI, который определяет местоположение оконечной точки. Разрешенный синтаксис для этого URI зависит от связи протокола.

– `ResponseLocation` [Дополнительный]

  Дополнительно определяет другое местоположение, куда должны передаваться сообщения ответы, передаваемые как часть протокола или профиля. Разрешенный синтаксис для этого URI зависит от связи протокола.

Атрибут `ResponseLocation` используется для того, чтобы дать возможность определить другие оконечные точки для получения сообщений запроса и ответа, связанных с протоколом или профилем, но не как средство для распределение нагрузки или избыточности (для этой цели может быть введено несколько элементов этого типа). Когда роль содержит элемент этого типа, относящийся к протоколу или профилю, для которого применим один-единственный тип сообщения (запрос или ответ), атрибут `ResponseLocation` неприменим.

ПРИМЕЧАНИЕ (информативное). – PE41 (см. OASIS PE:2006) разъясняет вышеприведенный параграф, добавляя к вышеприведенному тексту следующее предложение:

> Если атрибут `ResponseLocation` пропущен, то можно предположить, что любые сообщения-ответы, связанные с протоколом или профилем, будут обработаны на URI, указанном в атрибуте `Location`.

В большинстве ситуаций, элементы этого типа появляются в схеме в неограниченных последовательностях. Это также должно позволить, чтобы протокол или профиль были предложены элементом на нескольких оконечных точках обычно с различными связями протокола, что позволяет потребителю метаданных выбирать оконечную точку, приемлемую для его потребностей. Многие оконечные точки также выполняют распределение нагрузки или восстановление после отказа "на стороне клиента", в частности для синхронных связей протокола.

Этот элемент также позволяет использовать произвольные элементы и атрибуты, определенные в не-SAML области имен. Любое такое содержание должно быть определено в области имен.

Приведенный далее фрагмент схемы определяет сложный тип **EndpointType**:

```
<complexType name="EndpointType">
 <sequence>
    <any namespace="##other" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
 </sequence>
 <attribute name="Binding" type="anyURI" use="required"/>
 <attribute name="Location" type="anyURI" use="required"/>
 <attribute name="ResponseLocation" type="anyURI" use="optional"/>
 <anyAttribute namespace="##other" processContents="lax"/>
</complexType>
```

#### 9.1.2.3 Сложный тип IndexedEndpointType

Сложный тип **IndexedEndpointType** расширяет тип **EndpointType**, добавляя пару атрибутов для того, чтобы обеспечить возможность нумерации оконечных точек, которые в противном случае были бы идентичными, таким образом их можно будет указывать в протокольных сообщениях. Он состоит из следующих дополнительных атрибутов:

– `index` [Требуемый]

  Требуемый атрибут, который назначает оконечной точке уникальное целочисленное значение, так, чтобы его можно было указывать в протокольном сообщении. Значение индекса должно быть уникальным только в пределах набора похожих элементов, находящихся внутри одного и того же элемента-родителя (т. е. они не должны быть уникальными во всем элементе).

– `isDefault` [Дополнительный]

  Дополнительный Булев атрибут, используемый для обозначения оконечной точки "по умолчанию" внутри нумерованного множества. Если он пропущен, то предполагается что его значение = false.

В любой такой последовательности похожих конечных точек, основанных на этом типе, оконечная точка "по умолчанию" – это первая такая оконечная точка, у которой атрибут `isDefault` установлен в значение `true`. Если не существует таких оконечных точек, то оконечная точка "по умолчанию" – это первая такая оконечная точка, у которой атрибут `isDefault` не установлен в значение `false`. Если не существует таких оконечных точек, то оконечная точка "по умолчанию" – это первый элемент последовательности.

ПРИМЕЧАНИЕ (информативное). – PE37 (см. OASIS PE:2006) предлагает ввести в вышеприведенный параграф разъяснение следующего содержания:

> В любой такой последовательности пронумерованных оконечных точек, имеющих общее имя элемента и общую область имен (т. е. все они являются экземплярами `<md:AssertionConsumerService>` в пределах роли), оконечная точка "по умолчанию" – это первая такая оконечная точка, у которой атрибут `isDefault` установлен в значение `true`. Если не существует таких оконечных точек, то оконечная точка "по умолчанию" – это первая такая оконечная точка, у которой атрибут `isDefault` не установлен в значение `false`. Если не существует таких оконечных точек, то оконечная точка "по умолчанию" – это первый элемент последовательности.

Приведенный далее фрагмент схемы определяет сложный тип **IndexedEndpointType**:

```
<complexType name="IndexedEndpointType">
 <complexContent>
    <extension base="md:EndpointType">
       <attribute name="index" type="unsignedShort" use="required"/>
       <attribute name="isDefault" type="boolean" use="optional"/>
    </extension>
 </complexContent>
</complexType>
```

#### 9.1.2.4 Сложный тип localizedNameType

Сложный тип **localizedNameType** сложный тип расширяет элемент, имеющий строчное значение, добавляя в него стандартный атрибут языка XML. Приведенный далее фрагмент схемы определяет сложный тип **localizedNameType**:

```
<complexType name="localizedNameType">
 <simpleContent>
    <extension base="string">
       <attribute ref="xml:lang" use="required"/>
    </extension>
 </simpleContent>
</complexType>
```

#### 9.1.2.5 Сложный тип localizedURIType

Сложный тип **localizedURIType** расширяет элемент, имеющий значение URI, добавляя в него стандартный атрибут языка XML.

Приведенный далее фрагмент схемы определяет сложный тип **localizedURIType**:

```
<complexType name="localizedURIType">
    <simpleContent>
    <extension base="anyURI">
       <attribute ref="xml:lang" use="required"/>
    </extension>
 </simpleContent>
</complexType>
```

### 9.1.3 Корневые элементы

Экземпляр метаданных SAML описывает либо один элемент или несколько элементов. В первом случае корневым элементом должен быть элемент `<EntityDescriptor>`. В последнем случае корневым элементом должен быть элемент `<EntitiesDescriptor>`.

#### 9.1.3.1 Элемент <EntitiesDescriptor>

Элемент `<EntitiesDescriptor>` содержит метаданные для дополнительно названной группы элементов SAML. Его сложный тип **EntitiesDescriptorType** содержит последовательность либо элементов `<EntityDescriptor>`, либо элементов `<EntitiesDescriptor>` или обе последовательности:

– `ID` [Дополнительный]

  Идентификатор элемента, уникальный для данного документа, обычно используется как эталонная точки в процессе подписания.

– `validUntil` [Дополнительный]

  Дополнительный атрибут указывает время истечения срока жизни метаданных, содержащихся в элементе и любых элементах, которые находятся внутри него.

– `cacheDuration` [Дополнительный]

  Дополнительный атрибут указывает максимальную продолжительность интервала времени, в течение которого потребитель должен записать метаданные, содержащиеся в элементе и любых элементах, которые находятся внутри него.

– `Name` [Дополнительный]

  Строчное имя, которое идентифицирует группу элементов SAML в контексте определенной системы.

– `<ds:Signature>` [Дополнительный]

  Подпись XML, которая аутентифицирует элемент и его содержание, как описано в разделе 8.

– `<Extensions>` [Дополнительный]

  Он содержит дополнительные метаданные расширения, которые согласованы между создателем и потребителем метаданных. Элементы расширения должны быть определены в не-SAML области имен.

– `<EntitiesDescriptor>` или `<EntityDescriptor>` [Один или несколько]

  Содержит метаданные для одного или нескольких элементов SAML, или вложенную группу дополнительных метаданных.

Когда этот элемент используется в качестве корневого элемента экземпляра метаданных, он должен содержать либо атрибут `validUntil`, либо атрибут `cacheDuration`. Рекомендуется, чтобы этот атрибут содержался только в одном корневом элементе экземпляра метаданных.

Приведенный далее фрагмент схемы определяет элемент `<EntitiesDescriptor>` и его сложный тип **EntitiesDescriptorType**:

```
<element name="EntitiesDescriptor" type="md:EntitiesDescriptorType"/>
<complexType name="EntitiesDescriptorType">
 <sequence>
    <element ref="ds:Signature" minOccurs="0"/>
    <element ref="md:Extensions" minOccurs="0"/>
    <choice minOccurs="1" maxOccurs="unbounded">
       <element ref="md:EntityDescriptor"/>
       <element ref="md:EntitiesDescriptor"/>
    </choice>
 </sequence>
 <attribute name="validUntil" type="dateTime" use="optional"/>
 <attribute name="cacheDuration" type="duration" use="optional"/>
 <attribute name="ID" type="ID" use="optional"/>
 <attribute name="Name" type="string" use="optional"/>
</complexType>
```

```
<element name="Extensions" type="md:ExtensionsType"/>
<complexType final="#all" name="ExtensionsType">
 <sequence>
    <any namespace="##other" processContents="lax" maxOccurs="unbounded"/>
 </sequence>
</complexType>
```

#### 9.1.3.2 Элемент <EntityDescriptor>

Элемент `<EntityDescriptor>` определяет метаданные для одного-единственного элемента SAML. Один элемент может играть различные роли по поддержке различных профилей. Настоящая Рекомендация непосредственно поддерживает следующие конкретные роли, а также абстрактный элемент `<RoleDescriptor>` для обеспечения возможности расширения:

- SSO провайдера идентификации;
- SSO провайдер услуг;
- ответственный орган аутентификации;
- ответственный орган атрибута;
- точка принятия стратегического решения;
- объединение.

Его сложный тип **EntityDescriptorType** состоит из следующих элементов и атрибутов:

– `entityID` [Требуемый]

Определяет уникальный идентификатор элемента SAML, метаданные которого описываются содержанием элемента.

– `ID` [Дополнительный]

Идентификатор элемента, уникальный для данного документа, обычно используется как эталонная точки в процессе подписания.

– `validUntil` [Дополнительный]

Дополнительный атрибут указывает время истечения срока жизни метаданных, содержащихся в элементе и любых элементах, которые находятся внутри него.

– `cacheDuration` [Дополнительный]

Дополнительный атрибут указывает максимальную продолжительность интервала времени, в течение которого потребитель должен записать метаданные, содержащиеся в элементе и любых элементах, которые находятся внутри него.

– `<ds:Signature>` [Дополнительный]

Подпись XML, которая аутентифицирует элемент и его содержание.

– `<Extensions>` [Дополнительный]

Он содержит дополнительные метаданные расширения, которые согласованы между создателем и потребителем метаданных. Элементы расширения должны быть определены в не-SAML области имен.

– `<RoleDescriptor>`, `<IDPSSODescriptor>`, `<SPSSODescriptor>`, `<AuthnAuthorityDescriptor>`, `<AttributeAuthorityDescriptor>`, `<PDPDescriptor>` [Один или несколько]; или

– `<AffiliationDescriptor>` [Требуемый]

Обычно содержанием этого элемента является либо последовательность из одного или нескольких элементов дескриптора роли, либо специальный дескриптор, который определяет объединение.

– `<Organization>` [Дополнительный]

Дополнительный элемент, идентифицирующий организацию, ответственную за элемент SAML, описанный данным элементом.

– `<ContactPerson>` [Ноль или несколько]

Дополнительная последовательность элементов, идентифицирующих различные типы персонала, с которым следует поддерживать контакты.

– `<AdditionalMetadataLocation>` [Ноль или несколько]

Дополнительная последовательность местоположений, определенных в области имен, в которых имеются дополнительные метаданные для элемента SAML. Она может содержать метаданные в различных форматах или описания их соответствия с другими не-SAML рекомендациями.

Могут быть также включены произвольные атрибуты, определенные в не-SAML области имен.

Когда этот элемент используется в качестве корневого элемента экземпляра метаданных, он должен содержать либо атрибут `validUntil`, либо атрибут `cacheDuration`. Рекомендуется, чтобы этот атрибут содержался только в одном корневом элементе экземпляра метаданных.

Рекомендуется, чтобы в том случае, когда возникает несколько элементов дескрипторов роли одного и того же типа, они не использовали совместно перекрывающиеся значения `protocolSupportEnumeration`. Правила выбора элемента дескриптора роли из множества элементов дескрипторов роли одного типа, которые не используют совместно значение протокола `SupportEnumeration`, в настоящей Рекомендации не определяются, но могут быть определены профилями метаданных, возможно при помощи использования других атрибутов расширения.

Приведенный далее фрагмент схемы определяет элемент `<EntityDescriptor>` и его сложный тип **EntityDescriptorType**:

```
<element name="EntityDescriptor" type="md:EntityDescriptorType"/>
<complexType name="EntityDescriptorType">
 <sequence>
    <element ref="ds:Signature" minOccurs="0"/>
    <element ref="md:Extensions" minOccurs="0"/>
    <choice>
       <choice maxOccurs="unbounded">
          <element ref="md:RoleDescriptor"/>
          <element ref="md:IDPSSODescriptor"/>
          <element ref="md:SPSSODescriptor"/>
          <element ref="md:AuthnAuthorityDescriptor"/>
          <element ref="md:AttributeAuthorityDescriptor"/>
          <element ref="md:PDPDescriptor"/>
       </choice>
       <element ref="md:AffiliationDescriptor"/>
    </choice>
    <element ref="md:Organization" minOccurs="0"/>
    <element ref="md:ContactPerson" minOccurs="0" maxOccurs="unbounded"/>
    <element ref="md:AdditionalMetadataLocation" minOccurs="0"
maxOccurs="unbounded"/>
 </sequence>
 <attribute name="entityID" type="md:entityIDType" use="required"/>
 <attribute name="validUntil" type="dateTime" use="optional"/>
 <attribute name="cacheDuration" type="duration" use="optional"/>
 <attribute name="ID" type="ID" use="optional"/>
 <anyAttribute namespace="##other" processContents="lax"/>
</complexType>
```

#### 9.1.3.2.1 Элемент <Organization>

Элемент `<Organization>` определяет базовую информацию об организации, ответственной за элемент или роль SAML. Использовать этот элемент необязательно. Его содержание информативное по своей природе не соответствует непосредственно ни одному основному элементу или атрибуту SAML. Его сложный тип **OrganizationType** состоит из следующих элементов:

– `<Extensions>` [Дополнительный]

  Он содержит дополнительные расширения метаданных, которые согласованы между создателем и потребителем метаданных. Расширения не должны включать в себя глобальных (не определенных в области имен) элементов, или элементов, определенных в области имен SAML внутри этого элемента.

– `<OrganizationName>` [Один или несколько]

  Одно или несколько названий, определенных в используемом языке, которые могут быть пригодными для прочтения человеком, а могут таковыми и не быть.

– `<OrganizationDisplayName>` [Один или несколько]

  Одно или несколько названий, определенных в используемом языке, которые пригодны для прочтения человеком.

– `<OrganizationURL>` [Один или несколько]

  Один или несколько URI, определенных в используемом языке, которые определяют местоположение, куда следует направлять пользователя за дополнительной информацией. Определитель языка указывает содержание материала в определенном местоположении.

Могут быть также включены произвольные атрибуты, определенные в не-SAML области имен.

Приведенный далее фрагмент схемы определяет элемент <Organization> и его сложный тип **OrganizationType**:

```
<element name="Organization" type="md:OrganizationType"/>
<complexType name="OrganizationType">
 <sequence>
    <element ref="md:Extensions" minOccurs="0"/>
    <element ref="md:OrganizationName" maxOccurs="unbounded"/>
    <element ref="md:OrganizationDisplayName" maxOccurs="unbounded"/>
    <element ref="md:OrganizationURL" maxOccurs="unbounded"/>
 </sequence>
 <anyAttribute namespace="##other" processContents="lax"/>
</complexType>
<element name="OrganizationName" type="md:localizedNameType"/>
<element name="OrganizationDisplayName" type="md:localizedNameType"/>
<element name="OrganizationURL" type="md:localizedURIType"/>
```

#### 9.1.3.2.2 Элемент <ContactPerson>

Элемент <ContactPerson> определяет базовую контактную информацию о человеке, ответственном до некоторого предела за элемент или роль SAML. Использовать этот элемент необязательно. Его содержание информативное по своей природе не соответствует непосредственно ни одному основному элементу или атрибуту SAML. Его сложный тип **ContactType** состоит из следующих элементов и атрибутов:

– contactType [Требуемый]

  Определяет тип контакта с использованием нумерации типа **ContactTypeType**. Возможными значениями являются – техническая поддержка, административные функции, биллинг и другие.

– <Extensions> [Дополнительный]

  Он содержит дополнительные расширения метаданных, которые согласованы между создателем и потребителем метаданных. Элементы расширения должны быть определены в не-SAML области имен.

– <Company> [Дополнительный]

  Дополнительный строчный элемент, который определяет название компании контактного лица.

– <GivenName> [Дополнительный]

  Дополнительный строчный элемент, который определяет имя контактного лица.

– <SurName> [Дополнительный]

  Дополнительный строчный элемент, который определяет фамилию контактного лица.

– <EmailAddress> [Ноль или несколько]

  Ноль или несколько содержащих информацию об адресе электронной почты URI, представляющие адрес электронной почты контактного лица.

– <TelephoneNumber> [Ноль или несколько]

  Ноль или несколько строчных элементов, определяющих номер телефона контактного лица.

Могут быть также включены произвольные атрибуты, определенные в не-SAML области имен.

Приведенный далее фрагмент схемы определяет элемент <ContactPerson> и его сложный тип **ContactType**:

```
<element name="ContactPerson" type="md:ContactType"/>
<complexType name="ContactType">
 <sequence>
    <element ref="md:Extensions" minOccurs="0"/>
    <element ref="md:Company" minOccurs="0"/>
    <element ref="md:GivenName" minOccurs="0"/>
    <element ref="md:SurName" minOccurs="0"/>
    <element ref="md:EmailAddress" minOccurs="0" maxOccurs="unbounded"/>
    <element ref="md:TelephoneNumber" minOccurs="0" maxOccurs="unbounded"/>
 </sequence>
 <attribute name="contactType" type="md:ContactTypeType" use="required"/>
 <anyAttribute namespace="##other" processContents="lax"/>
```

```
</complexType>
<element name="Company" type="string"/>
<element name="GivenName" type="string"/>
<element name="SurName" type="string"/>
<element name="EmailAddress" type="anyURI"/>
<element name="TelephoneNumber" type="string"/>
<simpleType name="ContactTypeType">
 <restriction base="string">
    <enumeration value="technical"/>
    <enumeration value="support"/>
    <enumeration value="administrative"/>
    <enumeration value="billing"/>
    <enumeration value="other"/>
 </restriction>
</simpleType>
```

#### 9.1.3.2.3 Элемент <AdditionalMetadataLocation>

Элемент `<AdditionalMetadataLocation>` – это URI, определенный в области имен, который определяет, где смогут существовать дополнительные метаданные XML для элемента SAML. Его сложный тип **AdditionalMetadataLocationType** расширяет тип **anyURI**, добавляя атрибут области имен (который тоже имеет тип **anyURI**). Этот требуемый атрибут должен содержать область имен XML корневого элемента экземпляра данного документа, находящегося в указанном местоположении.

Приведенный далее фрагмент схемы определяет элемент `<AdditionalMetadataLocation>` и его сложный тип **AdditionalMetadataLocationType**:

```
<element name="AdditionalMetadataLocation"
type="md:AdditionalMetadataLocationType"/>
<complexType name="AdditionalMetadataLocationType">
 <simpleContent>
    <extension base="anyURI">
       <attribute name="namespace" type="anyURI" use="required"/>
    </extension>
 </simpleContent>
</complexType>
```

### 9.1.4 Элементы дескриптора роли

Элементы, описанные в настоящем разделе, образуют огромный блок компонентов метаданных эксплуатационной поддержки. Каждый элемент (за исключением абстрактного) определяет определенный набор эксплуатационных характеристик при поддержке профилей SAML.

#### 9.1.4.1 Элемент <RoleDescriptor>

Элемент `<RoleDescriptor>` – это абстрактная точка расширения, которая содержит общую описательную информацию, предназначенную для обеспечения однообразия обработки для различных ролей. Новые роли могут быть определены путем расширения абстрактного сложного типа **RoleDescriptorType**, который содержит следующие элементы и атрибуты:

– `ID` [Дополнительный]

  Идентификатор элемента, уникальный для данного документа, обычно используется как эталонная точки в процессе подписания.

– `validUntil` [Дополнительный]

  Дополнительный атрибут указывает время истечения срока жизни метаданных, содержащихся в элементе и любых элементах, которые находятся внутри него.

– `cacheDuration` [Дополнительный]

  Дополнительный атрибут указывает максимальную продолжительность интервала времени, в течение которого потребитель должен записать метаданные, содержащиеся в элементе и любых элементах, которые находятся внутри него.

– `protocolSupportEnumeration` [Требуемый]

  Ограниченный пробелами набор URI, которые идентифицируют набор спецификаций протокола, поддерживаемых элементом роли. Для элементов языка SAML V2.0, этот набор должен содержать URI области имен протокола SAML, `urn:oasis:names:tc:SAML:2.0:protocol`. Последующие Рекомендации по языку SAML могут использовать тот же URI области имен, но должны предусмотреть альтернативные идентификаторы "поддержки протокола" для того, чтобы обеспечить узнаваемость при необходимости.

– `errorURL` [Дополнительный]

  Дополнительный атрибут URI, который определяет место, куда следует направлять пользователя для решения проблем и за дополнительной поддержкой, связанной с его ролью.

– `<ds:Signature>` [Дополнительный]

  Подпись XML, которая аутентифицирует элемент и его содержание.

– `<Extensions>` [Дополнительный]

  Он содержит дополнительные метаданные расширения, которые согласованы между создателем и потребителем метаданных. Элементы расширения должны быть определены в не-SAML области имен.

– `<KeyDescriptor>` [Ноль или несколько]

  Дополнительная последовательность элементов, которая содержит информацию о ключах шифрования, которые использует элемент, когда исполняет эту роль.

– `<Organization>` [Дополнительный]

  Дополнительный элемент определяет организацию, связанную с этой ролью. Идентичен элементу, используемому внутри элемента `<EntityDescriptor>`.

– `<ContactPerson>` [Ноль или несколько]

  Дополнительная последовательность элементов, определяющая контактную информацию, связанную с этой ролью. Идентичен элементу, используемому внутри элемента `<EntityDescriptor>`.

Могут быть также включены произвольные атрибуты, определенные в не-SAML области имен.

Приведенный далее фрагмент схемы определяет элемент `<RoleDescriptor>` и его сложный тип **RoleDescriptorType**:

```
<element name="RoleDescriptor" type="md:RoleDescriptorType"/>
<complexType name="RoleDescriptorType" abstract="true">
 <sequence>
    <element ref="ds:Signature" minOccurs="0"/>
    <element ref="md:Extensions" minOccurs="0"/>
    <element ref="md:KeyDescriptor" minOccurs="0" maxOccurs="unbounded"/>
    <element ref="md:Organization" minOccurs="0"/>
    <element ref="md:ContactPerson" minOccurs="0" maxOccurs="unbounded"/>
 </sequence>
 <attribute name="ID" type="ID" use="optional"/>
 <attribute name="validUntil" type="dateTime" use="optional"/>
 <attribute name="cacheDuration" type="duration" use="optional"/>
 <attribute name="protocolSupportEnumeration" type="md:anyURIListType"
use="required"/>
 <attribute name="errorURL" type="anyURI" use="optional"/>
 <anyAttribute namespace="##other" processContents="lax"/>
</complexType>
<simpleType name="anyURIListType">
 <list itemType="anyURI"/>
</simpleType>
```

#### 9.1.4.1.1 Элемент <KeyDescriptor>

Элемент `<KeyDescriptor>` содержит информацию о ключе(ах) шифрования, которые использует элемент для подписания данных или получения зашифрованных ключей, вместе с дополнительными деталями шифрования. Его сложный тип **KeyDescriptorType** состоит из следующих элементов и атрибутов:

– `use` [Дополнительный]

  Дополнительный атрибут, определяющий цель применения описываемого ключа. Его значения берутся из нумерации типа **KeyTypes**, и состоят из значений `encryption` и `signing`.

– `<ds:KeyInfo>` [Требуемый]

  Дополнительный элемент, который прямо или косвенно идентифицирует ключ. Дополнительные подробности об использовании этого элемента содержатся в Правилах подписи XML Консорциума W3.

– `<EncryptionMethod>` [Ноль или несколько]

  Дополнительный элемент, определяющий алгоритм и зависящие от алгоритма установки, поддерживаемые элементом. Точное содержание меняется на основе поддерживаемого алгоритма. Определение сложного типа этого элемента **xenc:EncryptionMethodType** содержится в Правилах шифрования W3C.

Приведенный далее фрагмент схемы определяет элемент <KeyDescriptor> и его сложный тип **KeyDescriptorType**:

```
<element name="KeyDescriptor" type="md:KeyDescriptorType"/>
<complexType name="KeyDescriptorType">
    <sequence>
        <element ref="ds:KeyInfo"/>
        <element ref="md:EncryptionMethod" minOccurs="0"
maxOccurs="unbounded"/>
    </sequence>
    <attribute name="use" type="md:KeyTypes" use="optional"/>
</complexType>
<simpleType name="KeyTypes">
    <restriction base="string">
        <enumeration value="encryption"/>
        <enumeration value="signing"/>
    </restriction>
</simpleType>
<element name="EncryptionMethod" type="xenc:EncryptionMethodType"/>
```

#### 9.1.4.2 Сложный тип SSODescriptorType

Абстрактный тип **SSODescriptorType** – это общий базовый тип для конкретных типов **SPSSODescriptorType** и **IDPSSODescriptorType**, описываемых в последующих разделах. Он расширяет **RoleDescriptorType**, добавляя элементы, отражающие профили, общие для провайдеров идентификации и провайдеров услуг, которые поддерживают SSO, и содержит следующие дополнительные элементы:

– <ArtifactResolutionService> [Ноль или несколько]

  Ноль или несколько элементов типа **IndexedEndpointType**, которые описывают пронумерованные оконечные точки, которые поддерживают Профиль выделения артефактов, определенный в разделе 12. Атрибут ResponseLocation должен быть пропущен.

– <SingleLogoutService> [Ноль или несколько]

  Ноль или несколько элементов типа **EndpointType**, которые описывают оконечные точки, которые поддерживают профили единого выхода из системы, определенные в разделе 12.

– <ManageNameIDService> [Ноль или несколько]

  Ноль или несколько элементов типа **EndpointType**, которые описывают оконечные точки, которые поддерживают профили управления идентификатором имени, определенные в разделе 12.

– <NameIDFormat> [Ноль или несколько]

  Ноль или несколько элементов типа **anyURI**, которые перечисляют форматы идентификатора имени, поддерживаемые этим элементом системы, выполняющим эту роль.

  Приведенный далее фрагмент схемы определяет сложный тип **SSODescriptorType**:

```
<complexType name="SSODescriptorType" abstract="true">
 <complexContent>
   <extension base="md:RoleDescriptorType">
      <sequence>
         <element ref="md:ArtifactResolutionService" minOccurs="0"
maxOccurs="unbounded"/>
         <element ref="md:SingleLogoutService" minOccurs="0"
maxOccurs="unbounded"/>
         <element ref="md:ManageNameIDService" minOccurs="0"
maxOccurs="unbounded"/>
         <element ref="md:NameIDFormat" minOccurs="0"
maxOccurs="unbounded"/>
      </sequence>
   </extension>
 </complexContent>
</complexType>
<element name="ArtifactResolutionService" type="md:IndexedEndpointType"/>
<element name="SingleLogoutService" type="md:EndpointType"/>
<element name="ManageNameIDService" type="md:EndpointType"/>
<element name="NameIDFormat" type="anyURI"/>
```

#### 9.1.4.3 Элемент <IDPSSODescriptor>

Элемент `<IDPSSODescriptor>` расширяет тип **SSODescriptorType**, добавляя содержание, отражающее профили, специфичные для провайдеров идентификации, поддерживающих SSO. Его сложный тип **IDPSSODescriptorType** содержит следующие дополнительные элементы и атрибуты:

– `WantAuthnRequestsSigned` [Дополнительный]

Дополнительный атрибут, который указывает требование, что сообщения `<samlp:AuthnRequest>`, принятые этим провайдером идентификации, должны быть подписаны. Если он пропущен, предполагается, что его значение = `false`.

– `<SingleSignOnService>` [Один или несколько]

Один или несколько элементов типа **EndpointType**, которые описывают оконечные точки, которые поддерживают профили протокола запроса аутентификации, определенные в разделе 12. Все провайдеры идентификации, по определению, поддерживают как минимум одну такую оконечную точку. Атрибут `ResponseLocation` должен быть пропущен.

– `<NameIDMappingService>` [Ноль или несколько]

Ноль или несколько элементов типа **EndpointType**, которые описывают оконечные точки, которые поддерживают профиль Преобразования идентификатора имени, определенный в разделе 12. Атрибут `ResponseLocation` должен быть пропущен.

– `<AssertionIDRequestService>` [Ноль или несколько]

Ноль или несколько элементов типа **EndpointType**, которые описывают оконечные точки, которые поддерживают профиль протокола запроса подтверждения или специальную связь URI для запросов подтверждения, определенный в разделе 10.

ПРИМЕЧАНИЕ 1 (информативное). – PE33 (см. OASIS PE:2006) предлагает заменить протокол запроса подтверждения протоколом вопроса/запроса.

– `<AttributeProfile>` [Ноль или несколько]

Ноль или несколько элементов типа **anyURI**, которые перечисляют профили атрибута, поддерживаемые этим провайдером идентификации.

– `<saml:Attribute>` [Ноль или несколько]

Ноль или несколько элементов, которые идентифицируют атрибуты SAML, поддерживаемые провайдером идентификации. Дополнительно могут быть включены специальные значения, указывающие, что поддерживаются только определенные значения, разрешенные определением атрибута. В этом контексте "поддержка" атрибута означает, что провайдер идентификации имеет возможность его включить в процессе доставки подтверждения во время процесса единой регистрации в системе.

ПРИМЕЧАНИЕ 2 (информативное). – PE7 (см. OASIS PE:2006) предлагает добавить следующий текст в конец вышеприведенного параграфа:

Атрибут `WantAuthnRequestsSigned` предназначен для указания провайдерам услуг, могут ли они ожидать, что неподписанное сообщение `<AuthnRequest>` будет принято в обработку провайдером идентификации. Провайдер идентификации не обязан отбрасывать неподписанные запросы, а провайдер услуг не обязан подписывать свои запросы, хотя он вполне может ожидать, что неподписанные запросы будут отброшены. В некоторых случаях, провайдер услуг может даже не знать, какой провайдер идентификации, в конце концов, получит его запросы и ответит на них, поэтому использование этого атрибута в таком случае не может быть запрещено. Кроме того, отметим, что конкретный метод подписания зависит от связи. Описанная в 10.2.4 связь HTTP с перенаправлением требует, чтобы подпись применялась к значениям в кодировке URL, а не помещалась внутри сообщения XML, хотя другие связи, как правило, позволяют помещать подпись внутри сообщения обычным образом.

Приведенный далее фрагмент схемы определяет элемент `<IDPSSODescriptor>` и его сложный тип **IDPSSODescriptorType**:

```
<element name="IDPSSODescriptor" type="md:IDPSSODescriptorType"/>
<complexType name="IDPSSODescriptorType">
 <complexContent>
    <extension base="md:SSODescriptorType">
       <sequence>
          <element ref="md:SingleSignOnService" maxOccurs="unbounded"/>
          <element ref="md:NameIDMappingService" minOccurs="0"
maxOccurs="unbounded"/>
          <element ref="md:AssertionIDRequestService" minOccurs="0"
maxOccurs="unbounded"/>
```

```
         <element ref="md:AttributeProfile" minOccurs="0"
maxOccurs="unbounded"/>
         <element ref="saml:Attribute" minOccurs="0" maxOccurs="unbounded"/>
      </sequence>
      <attribute name="WantAuthnRequestsSigned" type="boolean"
use="optional"/>
    </extension>
 </complexContent>
</complexType>
<element name="SingleSignOnService" type="md:EndpointType"/>
<element name="NameIDMappingService" type="md:EndpointType"/>
<element name="AssertionIDRequestService" type="md:EndpointType"/>
<element name="AttributeProfile" type="anyURI"/>
```

#### 9.1.4.4 Элемент <SPSSODescriptor>

Элемент `<SPSSODescriptor>` расширяет тип **SSODescriptorType**, добавляя содержание, отражающее профили, специфичные для провайдеров услуг. Его сложный тип **SPSSODescriptorType** содержит следующие дополнительные элементы и атрибуты:

– `AuthnRequestsSigned` [Дополнительный]

Дополнительный атрибут, который указывает, должны ли быть подписаны сообщения `<samlp:AuthnRequest>`, передаваемые этим провайдером услуг. Если он пропущен, предполагается, что его значение = `false`.

ПРИМЕЧАНИЕ 1 (информативное). – PE7 (см. OASIS PE:2006) предлагает добавить следующий текст в конец вышеприведенного параграфа:

Значение "false" (или отсутствие этого атрибута) не означает, что провайдер услуг никогда не будет подписывать свои запросы или, что подписанный запрос должен считаться ошибкой. Однако провайдер идентификации, который получает неподписанное сообщение `<samlp:AuthnRequest>` от провайдера услуг, чьи метаданные содержит этот атрибут со значением `true`, должен возвратить ответ SAML с указанием ошибки и не должен выполнять этот запрос.

– `WantAssertionsSigned` [Дополнительный]

Дополнительный атрибут, который указывает требование, что элементы `<saml:Assertion>`, принятые этим провайдером услуг, должны быть подписаны. Если он пропущен, предполагается, что его значение = `false`. Это требование является дополнением к любым требованиям относительно подписания, полученных в результате использования конкретной комбинации профиль/связь.

ПРИМЕЧАНИЕ 2 (информативное). – PE7 (см. OASIS PE:2006) предлагает добавить следующий текст в конец вышеприведенного параграфа:

Отметим, что общая подпись на уровне связи или протокола SAML не является достаточной для выполнения этого требования, например для подписания ответа `<samlp:Response>`, содержащего подтверждение(я) или соединения TLS.

– `<AssertionConsumerService>` [Один или несколько]

Один или несколько элементов, которые описывают пронумерованные оконечные точки, которые поддерживают профили протокола запроса аутентификации, определенного в настоящей Рекомендации. Все провайдеры услуг, по определению, поддерживают как минимум одну такую оконечную точку.

– `<AttributeConsumingService>` [Ноль или несколько]

Ноль или несколько элементов, которые описывают приложение или услугу, предоставляемые провайдером услуг, который должен или желает использовать атрибуты SAML.

Максиму один элемент `<AttributeConsumingService>` может иметь атрибут `isDefault`, установленный в значение `true`. Допускается, чтобы ни один из используемых элементов не содержал атрибута `isDefault`, установленного в значение `true`.

Приведенный далее фрагмент схемы определяет элемент `<SPSSODescriptor>` и его сложный тип **SPSSODescriptorType**:

```
<element name="SPSSODescriptor" type="md:SPSSODescriptorType"/>
<complexType name="SPSSODescriptorType">
 <complexContent>
    <extension base="md:SSODescriptorType">
       <sequence>
          <element ref="md:AssertionConsumerService" maxOccurs="unbounded"/>
          <element ref="md:AttributeConsumingService" minOccurs="0"
maxOccurs="unbounded"/>
```

```
      </sequence>
      <attribute name="AuthnRequestsSigned" type="boolean" use="optional"/>
      <attribute name="WantAssertionsSigned" type="boolean" use="optional"/>
    </extension>
 </complexContent>
</complexType>
<element name="AssertionConsumerService" type="md:IndexedEndpointType"/>
```

#### 9.1.4.4.1 Элемент <AttributeConsumingService>

Элемент `<AttributeConsumingService>` определяет конкретную услугу, предлагаемую провайдером услуг в понятиях атрибутов услуги, которые требуются или желательны для этой услуги. Его сложный тип **AttributeConsumingServiceType** содержит следующие элементы и атрибуты:

– `index` [Требуемый]

  Требуемый атрибут, который назначает элементу уникальное целочисленное значение так, чтобы его можно было указывать в протокольном сообщении.

– `isDefault` [Дополнительный]

  Идентифицирует услугу, поддерживаемую провайдером услуг "по умолчанию". Полезен, если конкретная услуга никаким другим образом не указана в контексте приложения. Если он пропущен, предполагается, что его значение = `false`.

– `<ServiceName>` [Один или несколько]

  Одно или несколько названий услуги, определенных в используемом языке.

– `<ServiceDescription>` [Ноль или несколько]

  Ноль или несколько строк на используемом языке, которые описывают услугу.

– `<RequestedAttribute>` [Один или несколько]

  Один или несколько элементов, определяющих атрибуты, требуемые или желательные для данной услуги.

Приведенный далее фрагмент схемы определяет элемент `<AttributeConsumingService>` и его сложный тип **AttributeConsumingServiceType**:

```
<element name="AttributeConsumingService"
type="md:AttributeConsumingServiceType"/>
<complexType name="AttributeConsumingServiceType">
 <sequence>
    <element ref="md:ServiceName" maxOccurs="unbounded"/>
    <element ref="md:ServiceDescription" minOccurs="0"
maxOccurs="unbounded"/>
    <element ref="md:RequestedAttribute" maxOccurs="unbounded"/>
 </sequence>
 <attribute name="index" type="unsignedShort" use="required"/>
 <attribute name="isDefault" type="boolean" use="optional"/>
</complexType>
<element name="ServiceName" type="md:localizedNameType"/>
<element name="ServiceDescription" type="md:localizedNameType"/>
```

#### 9.1.4.4.2 Элемент <RequestedAttribute>

Элемент `<RequestedAttribute>` определяет заинтересованность провайдера услуг в конкретном атрибуте SAML, включая дополнительные конкретные значения. Его сложный тип **RequestedAttributeType** расширяет тип **saml:AttributeType**, добавляя следующий атрибут:

– `isRequired` [Дополнительный]

  Дополнительный атрибут XML указывает, требует ли услуге для нормальной работы соответствующий атрибут SAML (в отличие от простого определения атрибута как полезного или желательного).

  Если включаются конкретные атрибуты `<saml:AttributeValue>`, то для услуги пригодны только совпадающие значения.

Приведенный далее фрагмент схемы определяет элемент <RequestedAttribute> и его сложный тип **RequestedAttributeType**:

```
<element name="RequestedAttribute" type="md:RequestedAttributeType"/>
<complexType name="RequestedAttributeType">
 <complexContent>
    <extension base="saml:AttributeType">
       <attribute name="isRequired" type="boolean" use="optional"/>
    </extension>
 </complexContent>
</complexType>
```

#### 9.1.4.5 Элемент <AuthnAuthorityDescriptor>

Элемент <AuthnAuthorityDescriptor> расширяет тип **RoleDescriptorType**, добавляя содержание, отражающее профили, специфичные для ответственных органов аутентификации, т. е. ответственных органов SAML, которые отвечают на сообщения <samlp:AuthnQuery>. Его сложный тип **AuthnAuthorityDescriptorType** содержит следующие дополнительные элементы:

– <AuthnQueryService> [Один или несколько]

  Один или несколько элементов типа **EndpointType**, описывающие оконечные точки, которые поддерживают профиль протокола запроса аутентификации, определенный в разделе 12. Все ответственные органы аутентификации, по определению, поддерживают как минимум одну такую оконечную точку.

– <AssertionIDRequestService> [Ноль или несколько]

  Ноль или несколько элементов типа **EndpointType**, описывающие оконечные точки, которые поддерживают профиль протокола запроса подтверждения, определенный в разделе 12, или специальную связь URI для запросов подтверждения, определенную в разделе 10.

– <NameIDFormat> [Ноль или несколько]

  Ноль или несколько элементов типа **anyURI**, которые перечисляют форматы идентификатора имени, поддерживаемые этим ответственным органом (возможные значения этого элемента приведены в 8.7.3).

Приведенный далее фрагмент схемы определяет элемент <AuthnAuthorityDescriptor> и его сложный тип **AuthnAuthorityDescriptorType**:

```
<element name="AuthnAuthorityDescriptor"
type="md:AuthnAuthorityDescriptorType"/>
<complexType name="AuthnAuthorityDescriptorType">
 <complexContent>
    <extension base="md:RoleDescriptorType">
       <sequence>
          <element ref="md:AuthnQueryService" maxOccurs="unbounded"/>
          <element ref="md:AssertionIDRequestService" minOccurs="0"
maxOccurs="unbounded"/>
          <element ref="md:NameIDFormat" minOccurs="0"
maxOccurs="unbounded"/>
       </sequence>
    </extension>
 </complexContent>
</complexType>
<element name="AuthnQueryService" type="md:EndpointType"/>
```

#### 9.1.4.6 Элемент <PDPDescriptor>

Элемент <PDPDescriptor> расширяет тип **RoleDescriptorType**, добавляя содержание, отражающее профили, специфичные для точек принятия решений, и ответственные органы SAML, которые отвечают на сообщения <samlp:AuthzDecisionQuery>. Его сложный тип **PDPDescriptorType** содержит следующие дополнительные элементы:

– <AuthzService> [Один или несколько]

  Один или несколько элементов типа **EndpointType**, которые описывают оконечные точки, которые поддерживают профиль протокола запроса решения об авторизации, определенный в разделе 12. Все точки принятия решений, по определению, поддерживают как минимум одну такую оконечную точку.

– `<AssertionIDRequestService>` [Ноль или несколько]

Ноль или несколько элементов типа **EndpointType**, которые описывают оконечные точки, которые поддерживают профиль протокола запроса подтверждения, определенный в разделе 12, или специальную связь URI для запросов подтверждения, определенную в разделе 10.

ПРИМЕЧАНИЕ (информативное). – PE33 (см. OASIS PE:2006) предлагает заменить протокол запроса подтверждения протоколом вопроса/запроса.

– `<NameIDFormat>` [Ноль или несколько]

Ноль или несколько элементов типа **anyURI**, которые перечисляют форматы идентификатора имени, поддерживаемые этим ответственным органом (Возможные значения этого элемента приведены в 8.7.3).

Приведенный далее фрагмент схемы определяет элемент `<PDPDescriptor>` и его сложный тип **PDPDescriptorType**:

```
<element name="PDPDescriptor" type="md:PDPDescriptorType"/>
<complexType name="PDPDescriptorType">
 <complexContent>
    <extension base="md:RoleDescriptorType">
       <sequence>
          <element ref="md:AuthzService" maxOccurs="unbounded"/>
          <element ref="md:AssertionIDRequestService" minOccurs="0"
maxOccurs="unbounded"/>
          <element ref="md:NameIDFormat" minOccurs="0"
maxOccurs="unbounded"/>
       </sequence>
    </extension>
 </complexContent>
</complexType>
<element name="AuthzService" type="md:EndpointType"/>
```

#### 9.1.4.7 Элемент <AttributeAuthorityDescriptor>

Элемент `<AttributeAuthorityDescriptor>` расширяет тип **RoleDescriptorType**, добавляя содержание, отражающее профили, специфичные для ответственных органов атрибута, ответственных органов SAML, которые отвечают на сообщения `<samlp:AuthnQuery>`. Его сложный тип **AttributeAuthorityDescriptorType** содержит следующие дополнительные элементы:

– `<AttributeService>` [Один или несколько]

Один или несколько элементов типа **EndpointType**, описывающие оконечные точки, которые поддерживают профиль протокола запроса атрибута, определенный в разделе 12. Все ответственные органы атрибута, по определению, поддерживают как минимум одну такую оконечную точку.

– `<AssertionIDRequestService>` [Ноль или несколько]

Ноль или несколько элементов типа **EndpointType**, описывающие оконечные точки, которые поддерживают профиль протокола запроса подтверждения, определенный в разделе 12, или специальную связь URI для запросов подтверждения, определенную в разделе 10.

ПРИМЕЧАНИЕ (информативное). – PE33 (см. OASIS PE:2006) предлагает заменить протокол запроса подтверждения протоколом вопроса/запроса.

– `<NameIDFormat>` [Ноль или несколько]

Ноль или несколько элементов типа **anyURI**, которые перечисляют форматы идентификатора имени, поддерживаемые этим ответственным органом (Возможные значения этого элемента приведены в 8.7.3).

– `<AttributeProfile>` [Ноль или несколько]

Ноль или несколько элементов типа **anyURI**, которые перечисляют профили атрибута, поддерживаемые этим ответственным органом (Возможные значения этого элемента приведены в 8.7.3).

– `<saml:Attribute>` [Ноль или несколько]

Ноль или несколько элементов, которые идентифицируют атрибуты SAML, поддерживаемые этим ответственным органом. Дополнительно могут быть включены специальные значения, указывающие на то, что поддерживаются только определенные значения, разрешенные определением атрибута.

Приведенный далее фрагмент схемы определяет элемент <AttributeAuthorityDescriptor> и его сложный тип **AttributeAuthorityDescriptorType**:

```
<element name="AttributeAuthorityDescriptor"
type="md:AttributeAuthorityDescriptorType"/>
<complexType name="AttributeAuthorityDescriptorType">
 <complexContent>
    <extension base="md:RoleDescriptorType">
       <sequence>
          <element ref="md:AttributeService" maxOccurs="unbounded"/>
          <element ref="md:AssertionIDRequestService" minOccurs="0"
maxOccurs="unbounded"/>
          <element ref="md:NameIDFormat" minOccurs="0"
maxOccurs="unbounded"/>
          <element ref="md:AttributeProfile" minOccurs="0"
maxOccurs="unbounded"/>
          <element ref="saml:Attribute" minOccurs="0" maxOccurs="unbounded"/>
       </sequence>
    </extension>
 </complexContent>
</complexType>
<element name="AttributeService" type="md:EndpointType"/>
```

### 9.1.5 Элемент <AffiliationDescriptor>

Элемент <AffiliationDescriptor> является альтернативой последовательности дескрипторов ролей, которая используется, когда дескриптор <EntityDescriptor> описывает объединение элементов SAML (обычно, провайдеров услуг), а не отдельный элемент. Элемент <AffiliationDescriptor> содержит сумму отдельных элементов, которые формируют объединение, вместе с общей информацией о самом этом объединении. Его сложный тип **AffiliationDescriptorType** содержит следующие элементы и атрибуты:

– affiliationOwnerID [Требуемый]

  Определяет уникальный идентификатор элемента, ответственного за объединение. Не предполагается, что его владелец сам будет членом этого объединения; если же он член, то его идентификатор также должен содержаться в элементе <AffiliateMember>.

– ID [Дополнительный]

  Идентификатор элемента, уникальный для данного документа, обычно используется как эталонная точки в процессе подписания.

– validUntil [Дополнительный]

  Дополнительный атрибут указывает время истечения срока жизни метаданных, содержащихся в элементе и любых элементах, которые находятся внутри него.

– cacheDuration [Дополнительный]

  Дополнительный атрибут указывает максимальную продолжительность интервала времени, в течение которого потребитель должен записать метаданные, содержащиеся в элементе и любых элементах, которые находятся внутри него.

– <ds:Signature> [Дополнительный]

  Подпись XML, которая аутентифицирует элемент и его содержание (см. раздел 8).

– <Extensions> [Дополнительный]

  Он содержит дополнительные метаданные расширения, которые согласованы между создателем и потребителем метаданных. Элементы расширения должны быть определены в не-SAML области имен.

– <AffiliateMember> [Один или несколько]

  Один или несколько элементов, перечисляющих членов объединения, указывая уникальный идентификатор каждого участника (см. также 8.7.3.6).

– <KeyDescriptor> [Ноль или несколько]

  Дополнительная последовательность элементов, которая содержит информацию о ключах шифрования, которые используются всем объединением в целом, отличается от ключей, используемых отдельными членами объединения, которые публикуются в метаданных для этих элементов.

Могут быть также включены произвольные атрибуты, определенные в не-SAML области имен.

Приведенный далее фрагмент схемы определяет элемент <AffiliationDescriptor> и его сложный тип **AffiliationDescriptorType**:

```
<element name="AffiliationDescriptor" type="md:AffiliationDescriptorType"/>
<complexType name="AffiliationDescriptorType">
 <sequence>
    <element ref="ds:Signature" minOccurs="0"/>
    <element ref="md:Extensions" minOccurs="0"/>
    <element ref="md:AffiliateMember" maxOccurs="unbounded"/>
    <element ref="md:KeyDescriptor" minOccurs="0" maxOccurs="unbounded"/>
 </sequence>
 <attribute name="affiliationOwnerID" type="md:entityIDType"
use="required"/>
 <attribute name="validUntil" type="dateTime" use="optional"/>
 <attribute name="cacheDuration" type="duration" use="optional"/>
 <attribute name="ID" type="ID" use="optional"/>
 <anyAttribute namespace="##other" processContents="lax"/>
</complexType>
<element name="AffiliateMember" type="md:entityIDType"/>
```

### 9.1.6 Примеры

Далее приведен пример метаданных для элемента системы SAML, действующего в качестве провайдера идентификации и ответственного органа атрибута. Подпись показана в виде поля для подстановки, без фактического содержания.

```
<EntityDescriptor xmlns="urn:oasis:names:tc:SAML:2.0:metadata"
 xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion"
 xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
 entityID="https://IdentityProvider.com/SAML">
 <ds:Signature>...</ds:Signature>
    <IDPSSODescriptor WantAuthnRequestsSigned="true"
    protocolSupportEnumeration="urn:oasis:names:tc:SAML:2.0:protocol">
        <KeyDescriptor use="signing">
            <ds:KeyInfo>
                <ds:KeyName>IdentityProvider.com SSO Key</ds:KeyName>
            </ds:KeyInfo>
        </KeyDescriptor>
        <ArtifactResolutionService isDefault="true" index="0"
       Binding="urn:oasis:names:tc:SAML:2.0:bindings:SOAP"
       Location="https://IdentityProvider.com/SAML/Artifact"/>
        <SingleLogoutService
       Binding="urn:oasis:names:tc:SAML:2.0:bindings:SOAP"
       Location="https://IdentityProvider.com/SAML/SLO/SOAP"/>
        <SingleLogoutService
       Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-Redirect"
       Location="https://IdentityProvider.com/SAML/SLO/Browser"
       ResponseLocation="https://IdentityProvider.com/SAML/SLO/Response"/>
        <NameIDFormat>
         urn:oasis:names:tc:SAML:1.1:nameid-format:X509SubjectName
        </NameIDFormat>
        <NameIDFormat>
         urn:oasis:names:tc:SAML:2.0:nameid-format:persistent
        </NameIDFormat>
        <NameIDFormat>
         urn:oasis:names:tc:SAML:2.0:nameid-format:transient
        </NameIDFormat>
        <SingleSignOnService
       Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-Redirect"
       Location="https://IdentityProvider.com/SAML/SSO/Browser"/>
        <SingleSignOnService
       Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST"
```

```
      Location="https://IdentityProvider.com/SAML/SSO/Browser"/>
        <saml:Attribute
      NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:uri"
      Name="urn:oid:1.3.6.1.4.1.5923.1.1.1.6"
      FriendlyName="eduPersonPrincipalName">
        </saml:Attribute>
        <saml:Attribute
      NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:uri"
      Name="urn:oid:1.3.6.1.4.1.5923.1.1.1.1"
      FriendlyName="eduPersonAffiliation">
            <saml:AttributeValue>member</saml:AttributeValue>
            <saml:AttributeValue>student</saml:AttributeValue>
            <saml:AttributeValue>faculty</saml:AttributeValue>
            <saml:AttributeValue>employee</saml:AttributeValue>
            <saml:AttributeValue>staff</saml:AttributeValue>
        </saml:Attribute>
    </IDPSSODescriptor>
    <AttributeAuthorityDescriptor
    protocolSupportEnumeration="urn:oasis:names:tc:SAML:2.0:protocol">
        <KeyDescriptor use="signing">
            <ds:KeyInfo>
                <ds:KeyName>IdentityProvider.com AA Key</ds:KeyName>
            </ds:KeyInfo>
        </KeyDescriptor>
        <AttributeService
      Binding="urn:oasis:names:tc:SAML:2.0:bindings:SOAP"
      Location="https://IdentityProvider.com/SAML/AA/SOAP"/>
        <AssertionIDRequestService
      Binding="urn:oasis:names:tc:SAML:2.0:bindings:URI"
      Location="https://IdentityProvider.com/SAML/AA/URI"/>
        <NameIDFormat>
         urn:oasis:names:tc:SAML:1.1:nameid-format:X509SubjectName
        </NameIDFormat>
        <NameIDFormat>
         urn:oasis:names:tc:SAML:2.0:nameid-format:persistent
        </NameIDFormat>
        <NameIDFormat>
         urn:oasis:names:tc:SAML:2.0:nameid-format:transient
        </NameIDFormat>
        <saml:Attribute
      NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:uri"
      Name="urn:oid:1.3.6.1.4.1.5923.1.1.1.6"
      FriendlyName="eduPersonPrincipalName">
        </saml:Attribute>
        <saml:Attribute
      NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:uri"
      Name="urn:oid:1.3.6.1.4.1.5923.1.1.1.1"
      FriendlyName="eduPersonAffiliation">
            <saml:AttributeValue>member</saml:AttributeValue>
            <saml:AttributeValue>student</saml:AttributeValue>
            <saml:AttributeValue>faculty</saml:AttributeValue>
            <saml:AttributeValue>employee</saml:AttributeValue>
            <saml:AttributeValue>staff</saml:AttributeValue>
        </saml:Attribute>
    </AttributeAuthorityDescriptor>
    <Organization>
        <OrganizationName xml:lang="en">Identity Providers R
US</OrganizationName>
        <OrganizationDisplayName xml:lang="en">
         Identity Providers R US, a Division of Lerxst Corp.
        </OrganizationDisplayName>
        <OrganizationURL
xml:lang="en">https://IdentityProvider.com</OrganizationURL>
    </Organization>
</EntityDescriptor>
```

Далее приведен пример метаданных для элемента системы SAML, действующего в качестве провайдера услуг. Подпись показана в виде поля для подстановки, без фактического содержания. В целях иллюстрации показана одна услуга, в которой не требуется, чтобы пользователь идентифицировал себя уникальным образом, но требуется, чтобы он разрешал доступ на базе атрибута типа роли.

```
<EntityDescriptor xmlns="urn:oasis:names:tc:SAML:2.0:metadata"
 xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion"
 xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
 entityID="https://ServiceProvider.com/SAML">
 <ds:Signature>...</ds:Signature>
    <SPSSODescriptor AuthnRequestsSigned="true"
    protocolSupportEnumeration="urn:oasis:names:tc:SAML:2.0:protocol">
        <KeyDescriptor use="signing">
            <ds:KeyInfo>
                <ds:KeyName>ServiceProvider.com SSO Key</ds:KeyName>
            </ds:KeyInfo>
        </KeyDescriptor>
        <KeyDescriptor use="encryption">
            <ds:KeyInfo>
                <ds:KeyName>ServiceProvider.com Encrypt Key</ds:KeyName>
            </ds:KeyInfo>
            <EncryptionMethod
 Algorithm="http://www.w3.org/2001/04/xmlenc#rsa-1_5"/>
        </KeyDescriptor>
        <SingleLogoutService
      Binding="urn:oasis:names:tc:SAML:2.0:bindings:SOAP"
      Location="https://ServiceProvider.com/SAML/SLO/SOAP"/>
        <SingleLogoutService
      Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-Redirect"
      Location="https://ServiceProvider.com/SAML/SLO/Browser"
      ResponseLocation="https://ServiceProvider.com/SAML/SLO/Response"/>
        <NameIDFormat>
         urn:oasis:names:tc:SAML:2.0:nameid-format:transient
        </NameIDFormat>
        <AssertionConsumerService isDefault="true" index="0"
      Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-Artifact"
      Location="https://ServiceProvider.com/SAML/SSO/Artifact"/>
        <AssertionConsumerService index="1"
      Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST"
      Location="https://ServiceProvider.com/SAML/SSO/POST"/>
        <AttributeConsumingService index="0">
            <ServiceName xml:lang="en">Academic Journals R US</ServiceName>
            <RequestedAttribute
         NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:uri"
         Name="urn:oid:1.3.6.1.4.1.5923.1.1.1.7"
         FriendlyName="eduPersonEntitlement">
                <saml:AttributeValue>
                 https://ServiceProvider.com/entitlements/123456789
                </saml:AttributeValue>
            </RequestedAttribute>
        </AttributeConsumingService>
    </SPSSODescriptor>
    <Organization>
        <OrganizationName xml:lang="en">Academic Journals R
US</OrganizationName>
        <OrganizationDisplayName xml:lang="en">
         Academic Journals R US, a Division of Dirk Corp.
        </OrganizationDisplayName>
        <OrganizationURL
xml:lang="en">https://ServiceProvider.com</OrganizationURL>
    </Organization>
</EntityDescriptor>
```

## 9.2 Обработка подписи

Различные элементы в экземпляре метаданных могут быть подписаны цифровой подписью (что указано включением элемента `<ds:Signature>`), цифровая подпись имеет следующие преимущества.

### 9.2.1 Целостность метаданных

Аутентификация метаданных доверенной подписывающей стороной.

Цифровая подпись требуется не всегда, например, если доверяющая сторона получает информацию непосредственно от элемента, ее публикующего (без промежуточных элементов) по безопасному каналу с элементом, аутентифицированным для доверяющей стороны другими средствами, отличными от цифровой подписи.

Существует множество различных способов "непосредственной" аутентификации и создания безопасного канала между двумя сторонами. Это список включает в себя TLS, HMAC, механизмы на основе паролей и т. д. Кроме того, применимые требования по безопасности зависят от взаимодействующих приложений.

Кроме того, элементы могут наследовать подписи охватывающих их родительских элементов, которые сами являются подписанными.

При отсутствии такого контекста, рекомендуется, чтобы был подписан как минимум корневой элемент экземпляра метаданных.

### 9.2.2 Профиль подписи XML

Спецификация подписи XML Консорциума W3C содержит общий синтаксис XML для подписания данных, которому присуща гибкость и многовариантность. В настоящем разделе подробно рассмотрены ограничения этих возможностей, так чтобы процессоры SAML не были вынуждены работать с обширными возможностями обработки подписи XML. Такое применение определяет особый вариант использования атрибутов типа **xs:ID**, представленных в корневых элементах, к которым могут применяться подписи. Все эти атрибуты вместе в настоящем разделе называются атрибутами идентификатора.

**1) Форматы и алгоритмы подписания**

Подпись XML имеет три пути создания подписи, которые связывают ее с подписью документа: обертывание, конвертная и отсоединенная.

Метаданные SAML должны использовать конвертные подписи при подписании элементов, определенных в настоящей Процессоры SAML должны поддерживать использование техники RSA для подписания и проверки достоверности для операций с открытым ключом в соответствии с алгоритмом, определенным в http://www.w3.org/2000/09/xmldsig#rsa-sha1.

**2) Ссылки**

Подписанные элементы метаданных должны представлять значение атрибута идентификаторов для подписываемого элемента. Элемент может быть, а может и не быть корневым элементом реального документа XML, содержащего подписанный элемент метаданных.

Подписи должны содержать одну-единственную ссылку `<ds:Reference>`, содержащую ссылку на URI идентификатора значение атрибута подписываемого элемента метаданных. Например, если значение атрибута идентификатора = "foo", то атрибут `URI` в элементе `<ds:Reference>` должен быть = "#foo".

Вследствие этого, подпись элемента метаданных должна применяться к содержанию подписанного элемента и всех содержащихся в нем дочерних элементов.

**3) Метод канонического назначения каналов**

Варианты реализации SAML должны использовать метод Эксклюзивного канонического назначения каналов (Exclusive Canonicalization), с комментариями или без них, в обоих случаях – в элементе `<ds:CanonicalizationMethod>` объекта `<ds:SignedInfo>`, или в виде алгоритма `<ds:Transform>`. Использование метод Эксклюзивного канонического назначения каналов гарантирует, что подписи, созданные для метаданных SAML, встроенных в контекст XML, могут быть проверены независимо от этого контекста.

**4) Преобразования**

Подписи в метаданных SAML не должны содержать преобразований, кроме преобразования конвертной подписи (с идентификатором http://www.w3.org/2000/09/xmldsig#enveloped-signature) или преобразований эксклюзивного канонического назначения каналов (с идентификатором http://www.w3.org/2001/10/xml-exc-c14n# или http://www.w3.org/2001/10/xml-exc-14n#WithComments).

Те, кто проверяет подписи, может отбросить подписи, которые содержат другие алгоритмы преобразования, как недействительные. Если те, кто проверяет подписи, это делают, они должны гарантировать, что никакая часть содержания сообщения SAML не будет выведена из-под действия подписи. Это может быть выполнено путем подписания внешнего соглашения о том, какие преобразования приемлемы, или путем ручного выполнения преобразований контекста и повторной проверки результата на соответствие тем же самым метаданным SAML.

**5) KeyInfo**

Правила подписи W3C определяют использование элемента `<ds:KeyInfo>`. SAML не требует использования `<ds:KeyInfo>`, но и не налагает никаких ограничений на его использование. Следовательно, `<ds:KeyInfo>` может отсутствовать.

### 9.3 Публикация и распознавание метаданных

В настоящей Рекомендации описывается два механизма для публикации элемента (и для определения пользователем их местоположения) документов метаданных: при помощи "хорошо известного местоположения" путем прямого указания уникального идентификатора прямо разыменованного элемента (URI называют по-разному – либо *entityID*, либо *providerID*), либо косвенно путем публикации местоположения метаданных в DNS. Разрешены, конечно, и другие внешние механизмы. Потребитель, который поддерживает оба подхода, прежде чем использовать механизм "хорошо известного местоположения" должен попытаться отыскать при помощи DNS.

В том случае, когда для получения документа требуется его транспортировка по сети, транспортировка должна быть защищена при помощи механизмов обеспечивающих аутентификацию сервера и защиту целостности. Например, распознание на основе HTTP должно быть защищено при помощи TLS, как определено в IETF RFC 2246 и дополнении IETF RFC 3546.

В настоящем разделе описываются различные механизмы, призванные содействовать установлению доверия к точности и законности метаданных, включая использование подписей XML, аутентификации сервера TLS и подписей DNS. Вне зависимости от используемого(ых) механизма(ов), доверяющие стороны должны иметь некоторые средства, при помощи которых они могли бы установить доверие к информации метаданных прежде, чем их использовать.

#### 9.3.1 Публикация и распознание метаданных при помощи "хорошо известного местоположения"

В последующих подразделах описывается публикация и распознание метаданных при помощи "хорошо известного местоположения".

##### 9.3.1.1 Публикация

Элементы могут публиковать свои документы с метаданными в хорошо известном местоположении, помещая документ в месте, обозначенном его уникальным идентификатором, который должен иметь форму URL (а не URN). Настоятельно рекомендуется, чтобы для этой цели использовались элементы URL `https`. Механизм распознания, поддерживаемый схемой URL (например, распознание HTTP 1.1 302), может использоваться, если документ не помещается непосредственно в указанное место. Если протокол публикации допускает идентификацию типов содержания на базе MIME, то содержание экземпляра метаданных должно иметь тип `application/samlmetadata+xml`.

Документ XML, представленный в хорошо известном местоположении, должен описывать метаданные только для элемента, определенного уникальным идентификатором (т. е. корневым элементом должен быть дескриптор `<EntityDescriptor>` с элементом `ID`, соответствующим местоположению). Если требуется описать другие элементы, то должен использоваться элемент `<AdditionalMetadataLocation>`. Следовательно, элемент `<EntitiesDescriptor>` не должен использоваться в документах, опубликованных с использованием этого механизма, поскольку группа элементов таким идентификатором не определяется.

##### 9.3.1.2 Распознавание

Если уникальным идентификатором элемента является URL, то потребители метаданных могут попытаться непосредственно распознать уникальный идентификатор элемента способом, присущим данной схеме, при помощи преобразования идентификатора.

#### 9.3.2 Публикация и распознание при помощи DNS

Для улучшения доступности документов метаданных и обеспечения дополнительного преобразования между уникальным идентификатором элемента и местоположением метаданных, элементы могут публиковать места размещения своих документов с метаданными в их соответствующей зоне DNS, как определено в IETF RFC 1034. Уникальный идентификатор элемента (URI) используется как исходные данные для этого процесса. Поскольку идентификаторы URI – это гибкие идентификаторы, методы публикации мест размещения и процесс распознавания определяются схемой идентификатора URI и полным квалификационным именем. Места размещения URI для метаданных могут быть затем определены при помощи запросов Записи ресурса (RR) NAPTR, как определено в IETF RFC 2914 и IETF RFC 3403.

Рекомендуется, чтобы элементы публиковали свои записи ресурса в файлах подписанной зоны, используя IETF RFC 2535, так чтобы доверяющие стороны могли установить достоверность опубликованного местоположения, ответственный орган данной зоны и целостность ответа DNS. Если подписи зоны DNS представлены, доверяющие стороны должны удостоверить подписи должным образом.

##### 9.3.2.1 Публикация

В настоящей Рекомендации используется запись ресурса NAPTR, описанная в IETF RFC 2915 и IETF RFC 3403. Знакомство с этими документами приветствуется.

Система динамического обнаружения ресурсов (DDDS) – это система общего назначения для получения информации, основанная на исходной строке, зависящей от приложения, и применения хорошо известных правил для преобразования этой строки, пока не будет достигнуто условие завершения, требующее заглянуть

в базу данных, определенную конкретным приложением или распознать URL, используя правила, определенные этим приложением. DDDS определяет конкретный тип записи ресурса DNS и записи NAPTR, для хранения информации в DNS необходимо применять правила DDDS.

Элементы могут публиковать отдельные URL в тех случаях, когда требуется распространить множество документов метаданных, тили, когда требуются различные документы метаданных из-за наличия множества доверенных взаимосвязей, которым необходимы отдельные данные о ключах, или когда интерфейсам услуг требуется отдельные объявления метаданных. Это можно выполнить, используя дополнительный элемент `<AdditionalMetadataLocation>`, или используя возможность регулярного (типового) выражения и множества полей определения услуги в самой записи ресурса NAPTR.

Если протокол публикации допускает идентификацию типов содержания на базе MIME, то содержание экземпляра метаданных должно иметь тип `application/samlmetadata+xml`.

Если уникальным идентификатором элемента является URN, то публикация местоположения соответствующих метаданных выполняется, как определено в документе IETF RFC 3404. В противном случае распознавание местоположения метаданных выполняется, как определено ниже.

Далее представлен определенный приложением профиль DDDS для распознавания метаданных SAML:

**1) Первое общеизвестное правило**

"Первое общеизвестное правило" выполнения распознавания метаданных SAML заключается в том, чтобы разобрать на составляющие уникальный идентификатор элемента и выделить полное квалификационное имя домена (частное выражение 3).

**2) Поле порядка**

Поле порядка указывает порядок обработки каждой возвращенной записи ресурса NAPTR. Публикующая сторона может представить несколько записей ресурса NAPTR, которые должны быть обработаны распознающим приложением в порядке, указанном в этом поле.

**3) Поле предпочтения**

Для завершающих записей ресурса NAPTR публикующая сторона указывает порядок предпочтений использования в распознающем приложении. Распознающее приложение может проигнорировать этот порядок в тех случаях, когда значение поля услуги не отвечает требованием распознающей стороны (например, запись ресурса возвращает протокол, который приложение не поддерживает).

**4) Поле флага**

При распознавании метаданных SAML дважды используется флаг "U", который является терминальным, и нулевое значение (предполагающее, что должны быть обработаны дополнительные записи ресурса). Флаг "U" указывает, что результатом применения правил является URI.

**5) Поле услуги**

Поле услуги, присущее SAML, как описано в следующем BNF, объявляет режимы, при помощи которых должны быть сделаны доступными экземпляры документа(ов):

```
servicefield = 1("PID2U" / "NID2U") "+" proto [*(":" class) *(":"
servicetype)]
proto = 1("https" / "uddi")
class = 1[ "entity" / "entitygroup" )
servicetype = 1(si / "spsso" / "idpsso" / "authn" / "authnauth" / "pdp" /
"attrauth" / alphanum )
si = "si" [":" alphanum] [":endpoint"]
alphanum = 1*32(ALPHA / DIGIT)
```

где:

- `servicefield` PID2U преобразует уникальный идентификатор элемента в URL метаданных;
- `servicefield` NID2U преобразует `<NameID>` клиента в URL метаданных;
- `proto` описывает протокол получения (`https` или `uddi`). В случае UDDI URL будет URL http(s), указывающим документ WSDL;
- `class` определяет, описывает ли указываемый документ метаданных отдельный элемент или несколько элементов. В последнем случае, указываемый документ должен содержать элемент, определенный исходным уникальным идентификатором, как член группы элементов внутри самого документа, например `<AffiliationDescriptor>` или `<EntitiesDescriptor>`;
- `servicetype` разрешает элементу опубликовать метаданные для различных ролей и услуг как отдельные документы. Распознающая сторона, которая встречает несколько объявлений `servicetype`, будет ссылаться на соответствующий URI, в зависимости от того, какая услуга требуется для работы (например, элемент, работающий и как провайдер идентификации, и как провайдер услуг, может публиковать метаданные для каждой роли в различных местах). Тип услуги `authn` представляет собой оконечную точку `<SingleSignOnService>`;

– `si` (с дополнительным компонентом оконечной точки) разрешает публикующей стороне опубликовать метаданные для экземпляра услуги либо непосредственно, либо, объявляя оконечную точку SOAP (используя `endpoint`).

Например:

– `PID2U+https:entity` представляет полный документ метаданных элемента, доступный через протокол https;

– `PID2U+uddi:entity:si:foo` представляет местоположение документа WSDL, который описывает экземпляр услуги "foo";

– `PID2U+https:entitygroup:idpsso` представляет метаданные для группы элементов, действующих как SSO провайдеров идентификации, членом которой является исходный документ;

– `NID2U+https:idp` представляет метаданные для SSO провайдера идентификации клиента.

**6) Поля regex и замены**

Ожидаемым результатом обработки исходной строки функцией regex должен быть достоверный адрес `https` URL или узла UDDI (документа WSDL).

#### 9.3.2.2 Примеры NAPTR

В настоящем разделе приведены некоторые примеры URL и адресов e-mail, которые могут использоваться элементами, которые поддерживают NAPTR (см. IETF RFC 2915).

**a) Примеры метаданных элемента NAPTR**

Элементы публикуют URL метаданных в следующем виде:

```
$ORIGIN provider.biz

;; order pref f service regexp or replacement

IN NAPTR 100 10 "U" PID2U+https:entity
  "!^.*$!https://host.provider.biz/some/directory/trust.xml!" ""
IN NAPTR 110 10 "U" PID2U+https: entity:trust
  "!^.*!https://foo.provider.biz:1443/mdtrust.xml!" ""
IN NAPTR 125 10 "U" PID2U+https:"
IN NAPTR 110 10 "U" PID2U+uddi:entity
 "!^.*$!https://this.uddi.node.provider.biz/libmd.wsdl" ""
```

**b) Примеры идентификаторов имени**

Работодатель клиента `example.int` использует провайдера идентификации, который может использоваться компанией обеспечения офиса для аутентификации авторизованных покупателей. Поставщик берет адрес электронной почты пользователя `buyer@example.int` в качестве исходных данных для процесса распознавания, и разбирает этот адрес электронной почты для выделения FQDN (`example.int`). Работодатель публикует следующую запись NAPTR в DNS `example.int`:

```
$ORIGIN example.int

IN NAPTR 100 10 "U" NID2U+https:authn
  "!^([^@]+)@(.*)$!https://serv.example.int:8000/cgi-bin/getmd?\1!" ""
IN NAPTR 100 10 "U" NID2U+https:idp
  "!^([^@]+)@(.*)$!https://auth.example.int/app/auth?\1" ""
```

#### 9.3.2.3 Распознавание

При распознавании метаданных для элемента через DNS, в качестве исходных данных для процесса распознавания используется уникальный идентификатор элемента, а не его реальное местоположение. Обработка выполняется следующим образом:

– если уникальным идентификатором является URN, обработку вести в соответствии с этапами распознавания, определенными в IETF RFC 3403;

– в противном случае разобрать идентификатор для получения полного квалификационного имени домена;

– несколько раз запросить DNS для записей ресурса NAPTR домена до тех пор, пока не будет возвращена завершающая запись ресурса;

– определить, какую запись ресурса использовать, на основе полей услуги, затем полей порядка, затем полей предпочтений результирующего множества;

– получить документ(ы) в установленном(ых) местоположении(ях), как требуется приложением.

Для инициации распознавания информации о расположении метаданных, в некоторых случаях потребуется разложить уникальный идентификатор элемента (выраженный в виде URI) на один или несколько элементарных элементов.

Для инициации процесса разложения должно использоваться следующее регулярное выражение (regex):

```
^([^:/?#]+:)?/*([^:/?#]*@)?(([^/?:#]*\.)*(([^/?#:\.]+)\.([^/?#:\.]+)))(:\d+)?([^?#
]*)(\?[^#]*)?(#.*)?$
     1            2           34            56             7             8      9
10        11
```

Частное выражение 3 должно привести к получению Полного квалификационного имени домена (FQDN), которое будет основой для получения сведений о местоположении метаданных в этой зоне.

Завершив разложение идентификатора, приложение выполняет запрос DNS результирующего домена (частное выражение 5) для записей ресурса NAPTR; следует ожидать получения одного или нескольких ответов. Из полученного в результате множества, приложение может исключить любое определение услуги, которое не имеет отношения к выполненным операциям запроса.

Приложения распознавания должны упорядочить множество результатов в соответствии с данными поля порядка, и могут упорядочить множество результатов, основываясь на множестве предпочтений. От распознающей стороны не требуется следовать порядку, указанному в поле предпочтений. Результирующая(ие) запись(и) ресурса NAPTR используются несколько раз (на основе флага порядка), пока не будет достигнута завершающая запись ресурса NAPTR.

Результатом будет корректное абсолютное значение URL, которое затем используется для получения документа метаданных.

#### 9.3.2.4 Кэширование местоположения метаданных

Кэширование местоположения не должно превышать TTL зоны DNS, откуда были получены данные местоположения. Распознающая сторона должна получить последнюю (по времени) копию данных местоположения по истечении TTL данной зоны.

Сторона, публикующая документы метаданных, при внесении изменений в данные о местоположении документов метаданных, должна тщательно учитывать TTL зоны. Если эти изменения должны быть внесены, то публикующая сторона должна либо сохранить документы и в старом и в новом местоположениях до того момента, пока она не будет уверена в том, что все подтверждающие распознающие стороны имеют данные о новом местоположении (например, изменение времени зоны + TTL), либо поместить в старом местоположении ответ "перенаправить", указывающий новое местоположение.

### 9.3.3 Пост-обработка метаданных

В последующих подразделах описывается пост-обработка метаданных.

#### 9.3.3.1 Кэширование экземпляра метаданных

Кэширование документа не должно превышать атрибута `validUntil` или `cacheDuration` элементов объекта. Если элементы метаданных имеют родительские элементы, которые имеют правила кэширования, родительский элемент имеет преимущество.

Для того чтобы корректно обработать атрибут `cacheDuration`, потребители должны сохранить данные о дате и времени получения документа.

Когда истек срок жизни документа или элемента, потребитель должен получить свежую копию, для чего может потребоваться обновить данные о местоположении(ях) документа. Потребители должны выполнять кэш-обработку документа в соответствии с IETF RFC 2616, 13 и могут запрашивать у PPE3 сервера последние обновления даты и времени. Публикующая сторона должна обеспечить приемлемую кэш-обработку, как описано в IETF RFC 2616, 10.3.5 (304 – Не изменен).

#### 9.3.3.2 Выполнение перенаправлений HTTPS

Публикующая сторона может создавать команду HTTP Перенаправить (301 – Перемещен постоянно, 302 или 307 Временно перенаправить), как определено в IETF RFC 2616, и для использования агентов должны направляться по указанному URL в ответе Перенаправить. Перенаправления должны быть сформированы в том же протоколе, что и исходный запрос.

#### 9.3.3.3 Обработка подписей XML и общая обработка доверений

Обработка метаданных предлагает несколько механизмов для переговоров относительно степени доверия как к самим метаданным, так и к элементам, описываемым этими метаданными:

- доверие, обусловленное подписью зоны DNS, откуда были получены сведения об URL местоположения метаданных, гарантирующее точность сведения о местоположении документа метаданных;
- доверие, обусловленное обработкой подписи самого документа метаданных, гарантирующее целостность документа XML;
- доверие, обусловленное аутентификацией сервером TSL URL местоположения метаданных, гарантирующее идентификацию стороны, публикующей метаданные.

Пост-обработка документа метаданных должна включать в себя обработку подписи на уровне документа XML и может включать один из двух других процессов. В частности, доверяющая сторона может принять решение доверить процесс разложения и выделения любому из перечисленных ответственных органов. Сторона, публикующая метаданные, должна использовать механизм обеспечения целостности документа и может использовать один из двух других профилей обработки для установления степени доверия к документу метаданных, регулируемой правилами реализации. Необходимо следовать следующим положениям:

**1) Обработка подписанных зон DNS**

Должны быть выполнена проверка подписи зоны DNS, если она представлена, как описано в IETF RFC 2535.

**2) Обработка подписанных документов и фрагментов**

Опубликованные документы метаданных должны быть подписаны, как описано в настоящей Рекомендации, либо при помощи сертификата, созданного для получателя документа, либо другой доверенной стороной. Публикующая сторона может признать подписи других сторон средством подтверждения доверия.

Потребители метаданных должны подтвердить подписи, когда они представлены, на документе метаданных как описано в настоящей Рекомендации.

**3) Обработка сервера аутентификации во время получения метаданных при помощи TLS**

Настоятельно рекомендуется, чтобы публикующая сторона использовала URL TLS; следовательно, потребители должны рассмотреть вопрос доверия, обусловленного создателем сертификата TLS. URL публикации не всегда будут размещены в домене объекта документа метаданных; следовательно, потребители не должны предполагать наличие сертификатов, объектом которых является рассматриваемый элемент, поскольку он может размещаться у другой доверенной стороны.

Поскольку основа для такого доверия может быть недоступна для кэшированного документа, то в таких условиях следует использовать другие механизмы.

## 10 Связи для SAML

В настоящем разделе определяются SAML протокольные связи для использования подтверждений и сообщений запрос-ответ языка SAML в протоколах и системах связи.

Преобразование процесса обмена сообщениями запрос-ответ SAML в стандартные протоколы связи или обмена сообщениями называется *протокольными связями* SAML (или просто *связями*). Событие преобразования процесса обмена сообщениями запрос-ответ SAML в конкретный протокол связи `<FOO>` называется *<FOO> связью SAML* или *связью SAML <FOO>*.

Например, связь SAML SOAP описывает, как процесса обмена сообщениями запрос и ответ SAML преобразуется в процесса обмена сообщениями SOAP.

Цель настоящей Рекомендации состоит в том, чтобы достаточно подробно определить множество выбранных связей с целью гарантировать такое положение дел, при котором независимо разработанное программное обеспечение, соответствующее SAML, могло взаимодействовать в условиях, когда используются стандартные протоколы связи или обмена сообщениями.

Если в спецификации не определено иного, под связью следует понимать поддержку передачи любого сообщения протокола SAML, созданного из типов **samlp:RequestAbstractType** и **samlp:StatusОтветType**. Далее, когда связь обозначает "запрос и ответ SAML", следует понимать, то она означает любые протокольные сообщения, созданные из этих типов.

В тексте настоящей Рекомендации используются следующие типографические условные обозначения: `<ns:Element>`, XMLAttribute, **Datatype**, OtherKeyword. В некоторых случаях, используются угловые скобки для обозначения неоконченных элементов, а не элементов XML; смысл будет понятен из контекста.

### 10.1 Руководство по определению дополнительных связей протокола

Настоящая Рекомендация определяет выбранное множество связей протокола, но в будущем, вероятно, будут разработаны другие связи. В настоящем разделе предложено руководство для третьих сторон желающих определить дополнительные связи. Далее приведен перечень контрольных вопросов, которые следует рассмотреть для каждой связи протокола:

- определить три участка, идентифицирующих информацию: URI, который уникальным образом идентифицирует связь протокола, почтовый или электронный адрес автора и ссылка на ранее определенные связи или профили, которые эта новая связь обновляет или отменяет;
- описать набор операций взаимодействия между сторонами, участвующими в связи. Любые ограничения относительно приложений, используемых каждой стороной, и протоколов, используемых при каждом взаимодействии, должны быть явно названы;
- определить стороны, участвующие в каждом взаимодействии, включая данные о том, сколько сторон участвует, и будут ли участвовать промежуточные элементы;
- определить способ аутентификации сторон, участвующих в каждом взаимодействии, включая данные о том, требуется ли аутентификация и каковы приемлемые типы аутентификации;

– определить уровень поддержки целостности сообщения, включая механизмы, используемые для обеспечения целостности сообщения;
– определить уровень поддержки конфиденциальности, включая данные о том, может ли третья сторона увидеть содержание сообщения и подтверждения SAML, требуется ли обеспечить конфиденциальность связи, и рекомендованные механизмы для достижения конфиденциальности;
– определить состояния ошибки, включая состояния ошибки для каждого участника, особенно тех, кто получает и обрабатывает подтверждения или сообщения SAML;
– определить аспекты безопасности, включая анализ угроз и описание контрмер;
– определить аспекты метаданных, так, чтобы поддержка связей, включающих определенный протокол связи или используемых в определенном профиле, могла быть организована эффективным и обеспечивающим взаимодействие способом.

## 10.2 Связи протокола

В последующих подразделах определяются связи протокола, которые определяются спецификацией, как часть стандарта SAML.

### 10.2.1 Общие соображения

В последующих разделах описываются характеристики всех связей протокола, определенных для языка SAML.

#### 10.2.1.1 Использование RelayState

Некоторые связи определяют механизм "RelayState" для сохранения и передачи информации о состоянии. Когда такой механизм используется при передаче сообщения-запроса в качестве первого этапа протокола SAML, он определяет требования по выбору и использованию связи, которая будет затем использована для передачи ответа. То есть если сообщение-запрос SAML сопровождается данными RelayState, то отвечающая сторона SAML должен возвратить свой ответ протокола SAML, используя связь, которая также поддерживает механизм RelayState, и она должна поместить точно те же данные RelayState, которые она получила с запросом, в соответствующий параметр RelayState ответа.

#### 10.2.1.2 Безопасность

Если не объявлено иного, эти утверждения безопасности применяются ко всем связям. Связи могут также сделать дополнительные утверждения относительно этих параметров безопасности.

**1) Использование TLS 1.0**

Если в спецификации не определено иного, при любом использование TLS 1.0 в связи SAML (IETF RFC 2246), сервера должны аутентифицировать себя для клиентов, используя сертификат X.509 v3. Клиент должен установить идентификацию сервера на основании содержания сертификата (обычно путем проверки поля DN объекта сертификата, атрибута `subjectAltName` и т. д.).

**2) Аутентификация источника данных**

Аутентификация и запрашивающей и отвечающей стороны SAML, связанных с данным сообщением, является дополнительной и зависит от условий использования. Для аутентификации источника данных могут использоваться механизмы аутентификации, доступные на уровне обмена сообщениями SOAP, или из опорного базового протокола (например, для многих связей – это протокол TLS или HTTP).

Аутентификация транспорта не будет удовлетворять требованиям сквозной аутентификации источника в тех связях, где сообщение протокола SAML проходит через промежуточный элемент – в этом случае рекомендуется аутентификация сообщения.

Язык SAML предлагает механизмы, позволяющие сторонам аутентифицировать друг друга, но, в дополнение к этому, в языке SAML могут использоваться другие механизмы аутентификации для обеспечения безопасности самого языка SAML.

**3) Целостность сообщения**

Целостность сообщения для запросов SAML и ответов SAML является дополнительной и зависит от условий использования. Для обеспечения целостности сообщения может использоваться уровень безопасности опорного базового или механизм уровне обмена сообщениями SOAP.

Целостность транспорта не будет удовлетворять требованиям сквозной целостности в тех связях, где сообщение протокола SAML проходит через промежуточный элемент – в этом случае рекомендуется целостность сообщения.

**4) Конфиденциальность сообщения**

Конфиденциальность сообщения для запросов SAML и ответов SAML является дополнительной и зависит от условий использования. Для обеспечения конфиденциальности сообщения может использоваться уровень безопасности опорного базового или механизм уровне обмена сообщениями SOAP.

Конфиденциальность транспорта не будет удовлетворять требованиям сквозной конфиденциальности в тех связях, где сообщение протокола SAML проходит через промежуточный элемент.

**5) Другие аспекты безопасности**

До своего создания, каждая комбинация механизмов обеспечения аутентификации, целостности сообщения и конфиденциальности должна быть проанализирована на уязвимость для конкретного протокола обмена и условий использования (подробнее см. Дополнение I). В Рекомендации IETF RFC 2617 описываются возможные атаки в условиях HTTP, когда используются базовые схемы аутентификации или схемы аутентификации, основанные на содержании сообщения. Особое внимание следует уделить влиянию на безопасность операций кэширования.

### 10.2.2 Связь SAML SOAP

SOAP – это облегченный протокол, предназначенный для обмена структурированной информацией в децентрализованной, распределенной среде. Он использует XML технологии для определения расширяемой среды передачи сообщений, формируя конструкции сообщений, обмен которыми может быть осуществлен при помощи множества опорных протоколов. Эта среда разработана так, что не зависеть от любой конкретной программной модели и других вариантов реализации определенной семантики. Двумя основными целями разработки SOAP являются простота и расширяемость. SOAP пытается достичь этих целей при помощи исключения из среды обмена сообщениями тех возможностей, которые частот используются в распределенных системах. Эти возможности включают в себя, но не ограничиваются этим "надежность", "безопасность", "корреляция", "маршрутизация" и "шаблоны обмена сообщениями" (MEP).

Сообщение SOAP – это главным образом односторонняя передача между узлами SOAP – от передатчика SOAP на приемник SOAP, возможно выполненная через один или несколько промежуточных узлов SOAP. Ожидается, что сообщения SOAP будут объедены приложениями с целью создания более сложных шаблонов взаимодействия от простого запрос/ответ до многосторонних прямых и обратных "разговорных" обменов.

SOAP определяет формат сообщения XML, которое содержит части заголовка и тела сообщения, что позволяет передавать данные и управляющую информацию. SOAP также определяет правила обработки, связанные с этим форматом, и связь HTTP для передачи сообщения SOAP.

Связь SAML SOAP определяет, как использовать SOAP для передачи и приема запросов и ответов SAML.

Так же, как и SAML, SOAP может использоваться на разнообразных опорных транспортах. Эта связь имеет аспекты, не зависящие от протокола, но также позволяет, при необходимости, использовать SOAP в среде HTTP (обязательно требуется реализовать).

#### 10.2.2.1 Требуемая информация

**Идентификация**: `urn:oasis:names:tc:SAML:2.0:bindings:SOAP`

**Контактная информация**: security-services-comment@lists.oasis-open.org

**Описание**: Приведено ниже.

**Обновления**: `urn:oasis:names:tc:SAML:1.0:bindings:SOAP-binding`

#### 10.2.2.2 Не зависящие от протокола аспекты связи SAML SOAP

В последующих подразделах определяются аспекты связей SAML SOAP, которые не зависят от опорного протокола, например HTTP, при помощи которого передаются сообщения SOAP. Эта связь поддерживает использование только SOAP 1.1.

##### 10.2.2.2.1 Базовая операция

Сообщение SOAP 1.1 состоит из трех элементов: формат, данные заголовка и тело сообщения. Элементы протокола запроса-ответа SAML должны быть вложены в тело сообщения SOAP.

Протокол SOAP 1.1 определяет также дополнительную систему кодирования данных. Эта система не используется внутри связи SAML SOAP. Это означает, что сообщения SAML могут передаваться с использованием SOAP без перекодирования из "стандартной" схемы SAML в схему, основанную на кодировании SOAP.

Модель системы, используемая для разговоров на языке SAML в системе SOAP, это простая модель запрос-ответ.

- Элемент системы, действующий как запрашивающая сторона SAML, передает элемент запроса SAML в теле сообщения SOAP на элемент системы, действующий как отвечающая сторона SAML. Запрашивающая сторона SAML не должна включать в сообщение SOAP более одного запроса SAML или включать какие-либо дополнительные элементы XML в тело SOAP.
- Отвечающая сторона SAML должна возвратить либо элемент ответа SAML в теле другого сообщения SOAP, либо создать сообщение SOAP об ошибке. Отвечающая сторона SAML не должна включать в сообщение SOAP более одного ответа SAML или включать какие-либо дополнительные элементы XML в тело SOAP. Если отвечающая сторона SAML не может, по какой-либо причине, обработать запрос SAML, она должна создать сообщение SOAP об ошибке. Коды ошибки SOAP не должны передаваться для ошибок внутри домена проблем языка SAML, например, неспособность найти схему расширения или служить сигналом о том, что объекту не разрешен доступ к ресурсу в запросе авторизации.

ПРИМЕЧАНИЕ (информативное). – PE19 (см. OASIS PE:2006) предлагает заменить вышеприведенный параграф следующим:

Отвечающая сторона SAML должна вернуть сообщение SOAP, содержащее либо элемент ответа SAML в теле сообщения либо ошибку SOAP. не должна включать в сообщение SOAP более одного ответа SAML или включать какие-либо дополнительные элементы XML в тело SOAP. Коды ошибки SOAP не должны передаваться для ошибок внутри домена проблем языка SAML, например, неспособность найти схему расширения или служить сигналом о том, что объекту не разрешен доступ к ресурсу запросе авторизации.

Получив ответ SAML в сообщении SOAP, запрашивающая сторона SAML не должна передавать отвечающей стороне SAML код ошибки или другие сообщения об ошибке. Поскольку форматом для обмена сообщениями является простой шаблон запрос-ответ, добавление дополнительных блоков, таких как условия ошибки, внесет в протокол ненужные усложнения.

Документ W3C SOAP ссылается на первый проект спецификации схемы XML включающий устаревшую область имен. Запрашивающие стороны SAML должны создавать документы SOAP, ссылающиеся только на финальную область имен схемы XML и. Отвечающие стороны SAML должны иметь возможность обработать как область имен схемы XML, используемую в SOAP 1.1 (см. W3C SOAP), так и финальную версию области имен схемы XML.

#### 10.2.2.2.2 Заголовки SOAP

Запрашивающая сторона SAML в разговоре SAML по протоколу SOAP может добавить к сообщению SOAP дополнительные заголовки. Эта связь не определяет никаких дополнительных заголовков SOAP.

ПРИМЕЧАНИЕ 1. – Причиной того, что другое заголовки должны быть разрешены, является то, что некоторое программное обеспечения и библиотеки SOAP могут добавлять заголовки в сообщение SOAP, и это не регулируется процессом SAML. Кроме того, некоторые заголовки могут потребоваться для опорных протоколов, в которых требуется маршрутизация сообщений, или механизмами обеспечения безопасности сообщений.

Отвечающая сторона SAML не должна требовать введения в сообщение SOAP никаких заголовков для того, чтобы обеспечить корректную обработку самого сообщения SAML, но может требовать введения дополнительных заголовков, которые разрешают проблемы маршрутизации или отвечают требованиям обеспечения безопасности сообщений.

ПРИМЕЧАНИЕ 2. – Смысл в том, что требование дополнительных заголовков приведет к фрагментации стандарта SAML и ограничит возможность взаимодействия.

### 10.2.2.3 Использование SOAP в системе HTTP

Процессор SAML, который требует соответствия со связью SAML SOAP, должен реализовать SAML в системе SOAP в системе HTTP. В настоящем разделе описываются определенные параметры использования SOAP в системе HTTP, включая заголовки HTTP, кэширование и сообщение об ошибках.

Связь HTTP для SOAP описывается в W3C SOAP, 6.0. Она требует использования заголовка `SOAPAction` в качестве части запроса SOAP HTTP. Отвечающая сторона SAML не должна зависеть от значения этого заголовка. Запрашивающая сторона SAML может установить значение заголовка `SOAPAction` следующим:

```
http://www.oasis-open.org/committees/security
```

#### 10.2.2.3.1 Заголовки HTTP

Запрашивающая сторона SAML в разговоре SAML по протоколу SOAP в системе HTTP может добавить к запросу HTTP дополнительные заголовки. Эта связь не определяет никаких дополнительных заголовков HTTP.

ПРИМЕЧАНИЕ 1. – Причиной того, что другое заголовки должны быть разрешены, является то, что некоторое программное обеспечения и библиотеки HTTP могут добавлять заголовки в сообщение HTTP, и это не регулируется процессом SAML. Кроме того, некоторые заголовки могут потребоваться для опорных протоколов, в которых требуется маршрутизация сообщений, или механизмами обеспечения безопасности сообщений.

Отвечающая сторона SAML не должна требовать введения в запрос HTTP никаких заголовков для того, чтобы обеспечить корректную обработку самого сообщения SAML, но может требовать введения дополнительных заголовков, которые разрешают проблемы маршрутизации или отвечают требованиям обеспечения безопасности сообщений.

ПРИМЕЧАНИЕ 2. – Смысл в том, что требование дополнительных заголовков приведет к фрагментации стандарта SAML и ограничит возможность взаимодействия.

#### 10.2.2.3.2 Кэширование

Прокси–элементы HTTP не должны кэшировать протокольные сообщения SAML. Для обеспечения этого должны выполняться следующие правила.

При использовании HTTP 1.1 запрашивающие стороны должны:

- включить поле заголовка `Cache-Control`, установленное в значение "`no-cache, no-store`";
- включить поле заголовка `Pragma`, установленное в значение "`no-cache`".

При использовании HTTP 1.1 отвечающие стороны должны:

- включить поле заголовка `Cache-Control`, установленное в значение "`no-cache, no-store, must-revalidate, private`";
- включить поле заголовка `Pragma`, установленное в значение "`no-cache`";
- не включать заголовок `Validator`, например, `Last-Modified` или `Etag`.

#### 10.2.2.3.3 Сообщение об ошибках

Отвечающая сторона SAML, которая отказывается выполнять обмен сообщениями с запрашивающей стороной SAML, должна вернуть ответ "`403 Forbidden`". В этом случае содержание тела сообщения HTTP не имеет значения.

Как описано в W3C SOAP, 6.2, в случае возникновения ошибки SOAP при обработке запроса SOAP HTTP сервер SOAP должен возвратить ответ "`500 Internal Server Error`" и включить в ответ сообщение SOAP, содержащее элемент SOAP `<SOAP-ENV:fault>`. Ошибка этого типа должна быть возвращена при появлении ошибок, связанных с SOAP до того, как управление будет передано на процессор SAML, или когда процессор SOAP сообщает о внутренней ошибке (например, неправильная область имен SOAP XML, не может быть найдено местоположение схемы SAML, процессор SAML генерирует сообщение об исключительной ситуации и т. д.).

ПРИМЕЧАНИЕ (информативное). – PE19 (см. [OASIS Document Errata]) предлагает заменить первое предложение в вышеприведенном параграфе следующим текстом:

> Как описано в W3C SOAP, 6.2, в случае возникновения ошибки SOAP при обработке запроса SOAP, SOAP сервер HTTP должен возвратить ответ "`500 Internal Server Error`" и включить в ответ сообщение SOAP, содержащее элемент SOAP `<SOAP-ENV:fault>`.

В случае возникновения ошибки обработки SAML SOAP сервер HTTP должен ответить, передав сообщение "`200 OK`" и включить в ответ SAML внутри тела сообщения SOAP элемент`<samlp:Status>`, определенный в языке SAML. Элемент `<samlp:Status>` не появляется в теле сообщения SOAP самостоятельно, а только внутри ответа SAML какого либо вида.

Более подробная информация об использования кодов состояния SAML, содержится в разделе "Подтверждения и протоколы SAML" настоящей Рекомендации.

#### 10.2.2.3.4 Аспекты метаданных

Поддержку связи SOAP следует обозначить, указав либо URL оконечной точки, в которую должны быть переданы запросы, содержащиеся в сообщениях SOAP для конкретного протокола или профиля, либо при помощи определения порта WSDL/оконечной точки.

#### 10.2.2.3.5 Пример обмена сообщениями SAML с использованием протокола SOAP по HTTP

Далее приводится пример запроса, который требует подтверждения от ответственного органа SAML, содержащего утверждение атрибута.

```
POST /SamlService HTTP/1.1
Host: www.example.com
Content-Type: text/xml
Content-Length: nnn
SOAPAction: http://www.oasis-open.org/committees/security
<SOAP-ENV:Envelope
    xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
    <SOAP-ENV:Body>
        <samlp:AttributeQuery xmlns:samlp:="…"
xmlns:saml="…" xmlns:ds="…" ID="_6c3a4f8b9c2d" Version="2.0"
IssueInstant="2004-03-27T08:41:00Z"
            <ds:Signature> … </ds:Signature>
            <saml:Subject>
            …
            </saml:Subject>
        </samlp:AttributeQuery>
    </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
Following is an example of the corresponding response, which supplies an
assertion containing the attribute statement as requested.
HTTP/1.1 200 OK
Content-Type: text/xml
Content-Length: nnnn
<SOAP-ENV:Envelope
    xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
    <SOAP-ENV:Body>
        <samlp:Response xmlns:samlp="…" xmlns:saml="…" xmlns:ds="…"
ID="_6c3a4f8b9c2d" Version="2.0" IssueInstant="2004-03-27T08:42:00Z">
            <saml:Issuer>https://www.example.com/SAML</saml:Issuer>
      <ds:Signature> … </ds:Signature>
            <Status>
              <StatusCode Value="…"/>
            </Status>
```

```
            <saml:Assertion>
                <saml:Subject>
                …
                </saml:Subject>
                <saml:AttributeStatement>
            …
                </saml:AttributeStatement>
             </saml:Assertion>
        </samlp:Response>
    </SOAP-Env:Body>
</SOAP-ENV:Envelope>
```

### 10.2.3 Обратная связь SOAP (PAOS)

Эта связь усиливает обратную связь HTTP для спецификации SOAP (см. PAOS:2003). Разработчики должны соответствовать общим правилам обработки, определенных в документе PAOS, в дополнении к правилам, определенным в настоящей Рекомендации. В случае возникновения конфликта нормативным документом следует считать документ альянса Liberty Alliance POAS:2003.

#### 10.2.3.1 Требуемая информация

**Идентификация**: `urn:oasis:names:tc:SAML:2.0:bindings:PAOS`

**Контактная информация**: security-services-comment@lists.oasis-open.org

**Описание**: Приводится ниже.

**Обновления**: Нет.

#### 10.2.3.2 Обзор

Обратная связь SOAP – это механизм, при помощи которого запрашивающая сторона HTTP может заявить о своей возможности действовать в качестве отвечающей стороны SOAP или промежуточного элемента SOAP для запрашивающей стороны SAML. Запрашивающая сторона HTTP способна поддерживать шаблон, согласно которому запрос SAML передается ей в формате SOAP внутри ответа HTTP от запрашивающей стороны SAML, а запрашивающая сторона HTTP отвечает, передавая ответ SAML в формате SOAP внутри последующего запроса HTTP. Этот шаблон обмена сообщениями поддерживает работу случая, определенного в профиле ECP SSO, в котором запрашивающая сторона HTTP является промежуточным элементом в процессе обмена сообщениями аутентификации.

#### 10.2.3.3 Обмен сообщениями

Связь PAOS предусматривает двухкомпонентные шаблоны обмена сообщениями:

1) запрашивающая сторона HTTP передает запрос HTTP запрашивающей стороне SAML. Запрашивающая сторона SAML отвечает, передавая ответ HTTP содержащий формат SOAP, содержащий сообщение-запрос SAML;
2) далее, запрашивающая сторона HTTP передает исходной запрашивающей стороне SAML запрос HTTP, содержащий формат SOAP, содержащий сообщение-ответ SAML. Запрашивающая сторона SAML отвечает, передавая ответ HTTP, возможно, в ответ на оригинальный запрос услуги (см. шаг 1).

Профиль ECP использует связь PAOS для выполнения аутентификации клиента для провайдера услуг до того, как услуга предоставлена. Это выполняется следующим образом, шаги показаны на рисунке 10-1.

1) Клиент запрашивает услугу, используя запрос HTTP.
2) Провайдер услуг отвечает, передавая запрос аутентификации SAML. Он передается с использованием запроса SOAP, переданного в ответе HTTP.
3) Клиент возвращает ответ SOAP, переносящий ответ аутентификации SAML. Он передается с использованием нового запроса HTTP.
4) Предполагая, что аутентификация и авторизация провайдера услуг прошла успешно, провайдер услуг может ответить на исходный запрос услуги ответом HTTP.

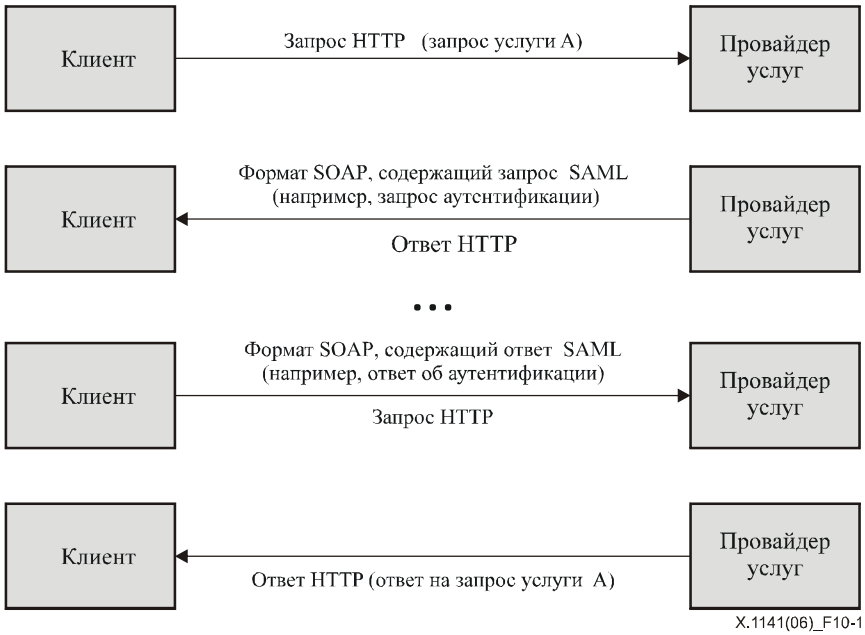


**Рисунок 10-1/X.1141 – Связь PAOS: обмен сообщениями**

Запрашивающая сторона HTTP объявляет о своей способности осуществить эту обратную связь SOAP в своих запросах HTTP с использованием заголовков HTTP, определенных в спецификации PAOS:2003. В частности:

- поле заголовка HTTP `Accept` должно указывать способность принять содержание типа "`application/vnd.paos+xml`";
- поле заголовка HTTP `PAOS` должно быть представлено и должно определять версию PAOS как минимум "`urn:liberty:paos:2003-08`".

ПРИМЕЧАНИЕ 1 (информативное). – PE21 (см. OASIS PE:2006) предлагает удалить слова "как минимум" из вышеприведенного текста.

Дополнительные заголовки PAOS, например значение услуги, могут быть определены профилями, которые используют связь PAOS. Запрашивающая сторона HTTP может добавить произвольные заголовки к запросу HTTP.

ПРИМЕЧАНИЕ 2. – Эта связь не определяет механизм RelayState. Следовательно, при необходимости, этот механизм должны определить конкретные профили, которые должны использовать эту связь. Для этой цели предполагается использовать заголовок SOAP.

В последующих разделах приводится более подробная информация об этих двух этапах обмена сообщениями.

#### 10.2.3.3.1 Запрос HTTP, запрос SAML в ответе SOAP

В ответ на произвольный запрос HTTP, отвечающая сторона HTTP может вернуть сообщение-запрос SAML, используя эту при помощи возврата формата SOAP 1.1 в ответе HTTP, содержащим одно-единственное сообщение-запрос SAML в теле сообщения SOAP без какого-либо дополнительного содержания в теле сообщения. Формат SOAP может содержать произвольные заголовки SOAP, определенные связью PAOS, профилями SAML или дополнительными Рекомендациями.

Когда сообщение-запрос SAML доставляется запрашивающей стороне HTTP, получателем, которого, в действительности, оно может быть другой элемент системы, где запрашивающая сторона HTTP действует в качестве промежуточного элемента, как определено в конкретных профилях.

#### 10.2.3.3.2 Ответ SAML в запросе SOAP, ответ HTTP

Когда запрашивающая сторона HTTP доставляет сообщение-ответ SAML тому получателю, которому оно предназначено, используя связь PAOS, она помещает его в виде одного-единственного элемента в теле сообщения SOAP в формате SOAP в запросе HTTP. Запрашивающая сторона HTTP может быть создателем ответа SAML, а может и не быть. Формат SOAP может содержать произвольные заголовки SOAP, определенные связью PAOS, профилями SAML или дополнительными Рекомендациями. Обмен сообщениями SAML считается завершенным, и ответ HTTP этой связью не определяется.

Профили могут налагать дополнительные ограничения на содержание HTTP ответов не-SOAP в ходе обмена сообщениями, относящимися к этой связи.

### 10.2.3.4 Кэширование

Прокси-элементы HTTP не должны кэшировать протокольные сообщения SAML. Для обеспечения этого должны выполняться следующие правила.

При использовании HTTP 1.1 запрашивающие стороны, передающие протокольные сообщения SAML, должны:

- включить поле заголовка `Cache-Control`, установленное в значение "`no-cache, no-store`";
- включить поле заголовка `Pragma`, установленное в значение "`no-cache`".

При использовании HTTP 1.1 отвечающие стороны, возвращающие протокольные сообщения SAML, должны:

- включить поле заголовка `Cache-Control`, установленное в значение "`no-cache, no-store, must-revalidate, private`";
- включить поле заголовка `Pragma`, установленное в значение "`no-cache`";
- не включать заголовок `Validator`, например, `Last-Modified` или `Etag`.

#### 10.2.3.5 Аспекты безопасности

Запрашивающая сторона HTTP в процессе связи PAOS может действовать в качестве промежуточного элемента SOAP, и когда это так, безопасность транспортного уровня для источника информации аутентификации, целостности и конфиденциальности может не отвечать требованиям по обеспечению сквозной безопасности. В этом случае рекомендуется обеспечить безопасность на уровне сообщения SOAP.

ПРИМЕЧАНИЕ (информативное). – PE31 (см. OASIS PE:2006) предлагает изменить слов рекомендуется словом РЕКОМЕНДУЕТСЯ.

#### 10.2.3.5.1 Сообщение об ошибках

Следует рассмотреть стандартные условные обозначения ошибок HTTP и SOAP. Об ошибках, которые появляются во время обработки, не должно сообщаться на уровни HTTP или SOAP, и они должны разрешаться с использованием сообщения-ответа SAML, содержащего элемент ошибки `<samlp:Status>`.

#### 10.2.3.5.2 Аспекты метаданных

Поддержку связи PAOS следует обозначить, указав URL оконечной точки, в которую должны быть переданы запросы HTTP и/или протокольные сообщения SAML, содержащиеся в форматах SOAP для конкретного протокола или профиля. Могут быть указаны либо одна оконечная точка, либо различные оконечные точки для запроса и ответа.

### 10.2.4 Связь перенаправления HTTP

Связь перенаправления HTTP определяет механизм, при помощи которого протокольные сообщения SAML могут передаваться в пределах параметров URL. Допустимая длина URL, теоретически, является бесконечной, но на практике она непредсказуемо ограничивается. Следовательно, требуются специальные коды для передачи сообщений XML по URL, и более объемное или более сложное содержание сообщений может быть передано с использованием связей HTTP POST или связей артефакта.

Эта связь может быть объединена вместе со связью HTTP POST (см. 10.2.5) или связью артефакта HTTP (см. 10.2.6) для передачи сообщений запросов и ответов в данном протоколе обмена, используя две различные связи.

Эта связь предполагает использование кодирования сообщение. Хотя определение этой связи включает определение одного конкретного способа кодирования сообщений, другие типы могут быть определены и могут использоваться.

#### 10.2.4.1 Требуемая информация

**Идентификация**: `urn:oasis:names:tc:SAML:2.0:bindings:HTTP-Redirect`

**Контактная информация**: security-services-comment@lists.oasis-open.org

**Описание**: Приводится ниже.

**Обновления**: Нет.

#### 10.2.4.2 Обзор

Связь перенаправления HTTP предназначена для случая, когда запрашивающая сторона SAML и отвечающая сторона SAML вынуждены связываться друг с другом, используя в качестве промежуточного элемента агент пользователя HTTP (определенный в IETF RFC 2616). Это может потребоваться, например, если связывающиеся стороны не имеют общего прямого пути для связи. Это может потребоваться также, если отвечающая сторона нуждается во взаимодействии с агентом пользователя для того, чтобы выполнять этот запрос, например, когда агент пользователя должен ее аутентифицировать.

Некоторые агенты пользователя HTTP могут иметь возможность исполнять более активную роль в протоколе обмена и могут поддерживать другие связи, использующие HTTP, например связи SOAP и обратная SOAP. Эта связь, кроме возможностей обычного веб-браузера, предполагает возможность передачи уведомлений.

#### 10.2.4.3 RelayState

Данные RelayState могут быть включены в протокольное сообщение SAML, передаваемое при помощи этой связи. Их длина не должна превышать 80 байтов, и их целостность должна быть защищена элементом, создающим сообщение, независимое от любых других защит, которые могут существовать или не существовать во время передачи сообщения. Подписание не реализуемо, учитывая ограничения объекта, но поскольку

существует угроза искажения этого значения третьей стороной, этот элемент должен обеспечить невозможность его искажения путем использования проверочной суммы, псевдослучайного значения или аналогичных средств.

ПРИМЕЧАНИЕ (информативное). – PE1 (см. OASIS PE:2006) утверждает, что последнее предложение в вышеприведенном параграфе должно иметь следующий вид:

> Данные RelayState могут быть включены вместе с протокольным сообщением SAML, передаваемым при помощи этой связи. Их длина значение не должна превышать 80 байтов, и их целостность должна быть защищена элементом, создающим сообщение, либо при помощи цифровой подписи (см. раздел 10) либо при помощи каких-либо независимых средств.

Если сообщение-запрос SAML является запросом SAML, то отвечающая сторона SAML должна возвратить свой протокольный ответ SAML, используя связь, которая также поддерживает механизм RelayState, и она должна поместить точно те же данные RelayState, которые она получила с запросом, в соответствующий параметра RelayState ответа.

Если в сообщение-запрос SAML не включено ни одного такого значения, или если сообщение-ответ SAML создается без соответствующего запроса, то отвечающая сторона SAML может включить данные RelayState, которые должны быть интерпретированы получателем на основе использования профиля или предварительного соглашения между сторонами.

#### 10.2.4.4 Кодирование сообщений

Для использования с этой связью сообщения кодируются кодировкой URL и передаются при помощи метода HTTP GET. Существуют множество возможных способов закодировать XML в URL, в зависимости от действующих ограничений. В настоящей Рекомендации определяется один из этих методов, что не препятствует использованию других. Оконечные точки связи, при необходимости, должны указывать с использованием метаданных, какое кодирование они поддерживают. Конкретное кодирование, когда оно определено, должно быть уникально указано при помощи URI. Если этого не требуется, то должна существовать возможность кодировать все возможные сообщения SAML с применением определенного набора правил, но эти правила должны явно указывать, какие сообщения или какое содержание может или не может быть закодировано этим способом.

Кодирование URL должно помещать сообщение целиком в строку запроса URL, и должно резервировать оставшуюся часть URL для оконечной точки получателя сообщения.

Параметр строки запроса с названием `SAMLEncoding` зарезервированы для определения используемого механизма кодирования. Если этот параметр пропущен, то предполагается, что его значение = `urn:oasis:names:tc:SAML:2.0:bindings:URL-Encoding:DEFLATE`.

Все оконечные точки, которые поддерживают эту связь, должны поддерживать кодирование DEFLATE, описанное далее.

**i) Кодирование DEFLATE**

**Идентификация**: `urn:oasis:names:tc:SAML:2.0:bindings:URL-Encoding:DEFLATE`

Протокольные сообщения SAML могут быть закодированы в сообщение URL при помощи метода компрессии DEFLATE (IETF RFC 1951). В этом методе кодирования, к XML версии исходного протокольного сообщения SAML должна применяться нижеследующая процедура.

1) любая подпись на протокольном сообщении SAML, включая сам элемент XML `<ds:Signature>`, должна быть удалена. Если содержание сообщения содержит другую подпись, например, подписанное подтверждение SAML, то эта вложенная подпись не удаляется. Однако длина такого сообщения после кодирования не даст возможности использовать этот механизм. Следовательно, протокольные сообщения SAML, в которых имеется подписанное содержание, не должны кодироваться с использованием этого механизма.
2) Затем ко всему остальному содержанию XML исходного протокольного сообщения SAML применяется механизм компрессии DEFLATE (определенный в документе IETF RFC 1951).
3) Компрессированные данные затем кодируются кодом base64 в соответствии с правилами, определенными в IETF RFC 2045. Символы перевода строки или другие пробелы должны быть удалены из результата.
4) Данные, кодированные кодом base64, затем кодируются в виде URL и добавляются к URL в виде параметра строки запроса, который должен иметь название `SAMLRequest` (если сообщением является запрос SAML) или `SAMLResponse` (если сообщением является ответ SAML).
5) Если данные RelayState должны сопровождать сообщение протокола SAML, то они должны быть кодированы в URL и помещены в параметр строки запроса, имеющий название `RelayState`.
6) Если исходное протокольное сообщение SAML было подписано с использованием цифровой подписи XML, то для определенных выше кодированных данных должна быть сформирована новая подпись с использованием перечисленных ниже правил.

Цифровые подписи XML не кодируются непосредственно в URL в соответствии с вышеприведенными правилами, из-за проблем с объемом данных. Если используемое протокольное сообщение SAML подписано подписью XML, сообщения кодированное в URL формате, должно быть подписано следующим образом:

1) должен быть введен идентификатор алгоритма подписи, как дополнительный параметр строки запроса, имеющий название `SigAlg`. Значением этого параметра должен быть URI, который идентифицирует алгоритм, используемый для подписи протокольное сообщение SAML, кодированное в формате URL, определенный в соответствии с подписью XML или какой-либо иной Рекомендацией, которая управляет этим алгоритмом;

2) для создания подписи создается строка, состоящая из объединения следующих параметров строки запроса `RelayState` (если представлен), `SigAlg` и `SAMLRequest` (или `SAMLResponse`) (каждый в виде URL-кода), одним из следующих способов (в порядке показанном ниже):

   a) `SAMLRequest=value&RelayState=value&SigAlg=value`
      `SAMLResponse=value&RelayState=value&SigAlg=value`

   b) результирующая строка байтов должна быть введена в алгоритм подписи. Любое другое содержание, имеющееся в исходной строке запроса, не включается и не подписывается;

   c) значение подписи должно быть закодировано с использованием кода base64 (см. IETF RFC 2045), причем все проблемы должны быть удалены, и введено как параметр строки запроса, имеющий название `Signature`. Некоторые символы в подписи, кодированной кодом base64, могут сами требовать кодирования URL до того, как добавляются;

   d) этим механизмом кодирования должны поддерживаться следующие алгоритмы подписи (см. Правила подписи W3C) и их представления в виде URI:

      - DSAwithSHA1 – http://www.w3.org/2000/09/xmldsig#dsa-sha1;
      - RSAwithSHA1 – http://www.w3.org/2000/09/xmldsig#rsa-sha1.

   ПРИМЕЧАНИЕ. – NIST (Национальный институт стандартов и технологии) сегодня приветствует использование SHA-256 (защищенный алгоритм хеширования с закодированными 256-битовыми ключами) вместо SHA-1.

При проверке подписей порядок параметров строки запроса в результирующем URL, который следует проверить, не предписывается этой связью. Параметры могут быть расположены в любом порядке. До проверки подписи, если она выполняется, доверяющая сторона должна гарантировать, что значения параметра, который следует проверить, расположены в порядке, требуемом правилами подписи, приведенными выше.

Кодирование URL не является канонически; т. е. существует множество законных методов закодировать данное значение. Следовательно, доверяющая сторона должна выполнить тап проверки, используя исходные значения в кодировке URL, которые она получила в строке запроса. Вовсе не достаточно просто заново закодировать параметры после их обработки программным обеспечением, поскольку результирующее кодирование может не совпадать с кодами того, кто подписал сообщения.

Если значения `RelayState` нет, то в расчете подписи весь параметр не должен учитываться (и не должен включаться как пустое имя параметра).

#### 10.2.4.5 Обмен сообщениями

Модель системы, используемая для соединений SAML при помощи этой связи, – это модель запрос – ответ, но эти сообщения передаются агенту пользователя в ответе HTTP и доставляются до получателя сообщения в запросе HTTP. Процессы взаимодействия процедуры взаимодействия HTTP до, между и после того, как выполнены эти обмены, в спецификации не определены. Как запрашивающая сторона SAML, так и отвечающая сторона SAML считаются отвечающими сторонами HTTP. Далее приведена последовательная диаграмма (рисунок 10-2), иллюстрирующая процесс обмена сообщениями.

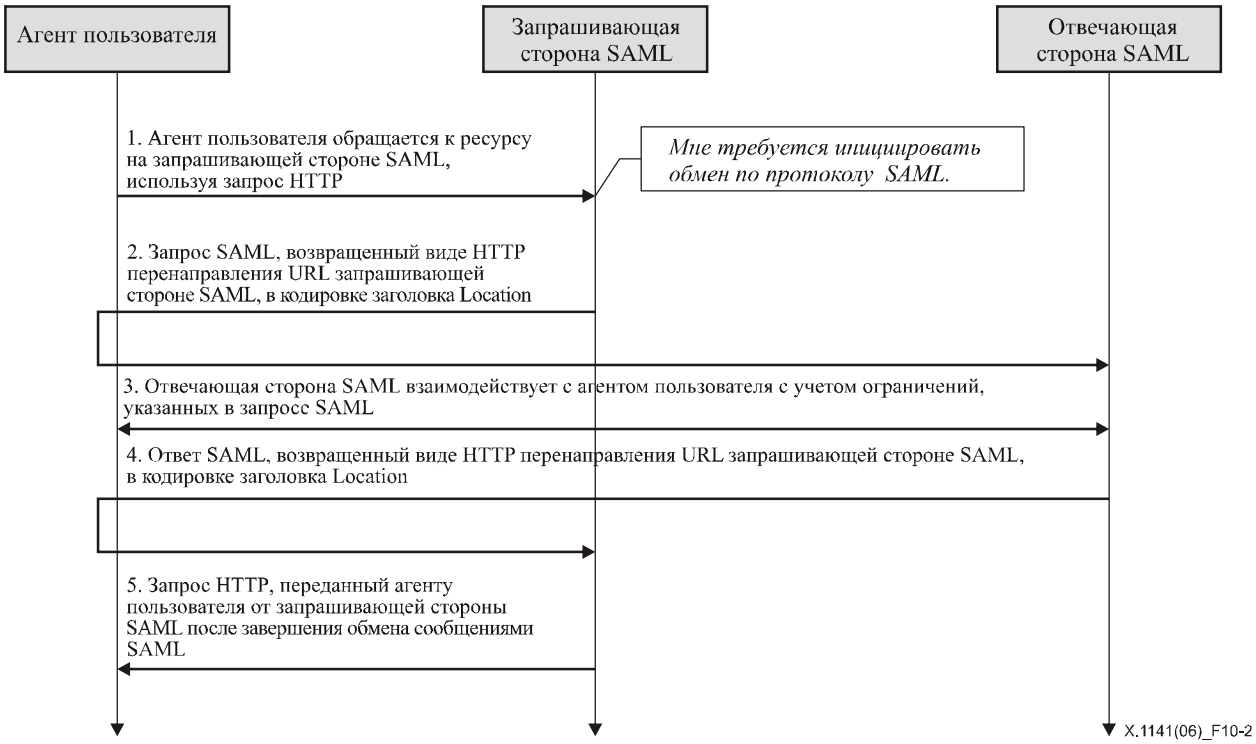


**Рисунок 10-2/X.1141 – Обмен сообщениями с перенаправлением HTTP**

1) Сначала агент пользователя передает элементу системы произвольный запрос HTTP. В ходе обработки запроса элемент системы принимает решение инициировать протокольный обмен SAML.
2) Элемент системы, действующий как запрашивающая сторона SAML, отвечает на запрос HTTP от агента пользователя на этапе 1, возвращая запрос SAML. Запрос SAML возвращается, закодированный в виде заголовка ответа HTTP Location, и статус HTTP должен быть либо 303, либо 302. Запрашивающая сторона SAML может включить в ответ HTTP дополнительное представление и содержание для того, чтобы помочь агенту пользователя передать сообщения, как определено в IETF RFC 2616. Агент пользователя доставляет отвечающей стороне SAML запрос SAML, создавая запрос HTTP GET.
3) Как правило, отвечающая сторона SAML может ответить на запрос SAML, сразу же возвратив ответ SAML, или может вернуть произвольной содержание для упрощения последующего взаимодействия с агентом пользователя, требуемого для выполнения этого запроса. Конкретные протоколы и профили могут содержать механизм, указывающий степень желания запрашивающей стороны разрешить этот вид взаимодействия (например, атрибут `IsPassive` в `<samlp:AuthnRequest>`).
4) В итоге, отвечающая сторона должна вернуть SAML ответ агенту пользователя, чтобы он вернул его запрашивающей стороне SAML. Ответ SAML возвращается таким же образом, как это было описано для запроса SAML в шаге 2.
5) После получения ответа SAML, запрашивающая сторона SAML возвращает произвольный ответ HTTP агенту пользователя.

#### 10.2.4.5.1 HTTP аспекты кэширования

Прокси-элементы HTTP и промежуточные элементы агента пользователя не должны кэшировать протокольные сообщения SAML. Для обеспечения этого, должны выполняться следующие правила.

Возвращая протокольные сообщения SAML с использованием HTTP 1.1, отвечающие стороны HTTP должны:

- включить поле заголовка `Cache-Control`, установленное в значение "`no-cache, no-store`";
- включить поле заголовка `Pragma`, установленное в значение "`no-cache`".

Не существует иных ограничений на использование заголовков HTTP.

#### 10.2.4.5.2 Аспекты безопасности

Существование промежуточного элемента агента пользователя означает, что запрашивающая сторона и отвечающая сторона не могут надеяться на транспортный уровень для обеспечения сквозной аутентификация, целостности и конфиденциальности. Сообщения в кодировке URL могут быть подписаны для обеспечения аутентификации и целостности источника, если метод кодирования определяет средства подписи.

Если сообщение подписано, то атрибут XML `Destination` в корневом элементе протокольного сообщения SAML должен содержать URL, куда отправитель приказывает агенту пользователя доставить сообщение. Получатель должен затем проверить, чтобы это значение соответствовало местоположению, из которого было получено сообщение.

Эта связь не должна использоваться, если содержание запроса или ответа не должно быть раскрыто промежуточному элементу агента пользователя. В противном случае требование конфиденциальности и запросов SAML, и ответов SAML является дополнительным и зависит от условий использования. Если конфиденциальность необходима, то должен использоваться TLS 1.0 для защиты сообщения при передаче между агентом пользователя и запрашивающей стороной или отвечающей стороной SAML.

Сообщения в кодировке URL могут быть раскрыты в различных файлах регистрации HTTP, а также в виде заголовка HTTP "Referrer".

До своего создания, каждая комбинация механизмов обеспечения аутентификации, целостности сообщения и конфиденциальности должна быть проанализирована на уязвимость для конкретного протокола обмена и условий использования (см. Дополнение I).

Как правило, эта связь надеется на аутентификации и защиту целостности на уровне протокола и при помощи подписи, и не поддерживает конфиденциальность сообщений, полученных от промежуточного элемента агента пользователя.

### 10.2.4.6 Сообщение об ошибках

Отвечающая сторона SAML, которая отказывается выполнять обмен сообщениями с запрашивающей стороной SAML, должна вернуть сообщение-ответ SAML со значением кода второго уровня `<samlp:StatusCode>` = `urn:oasis:names:tc:SAML:2.0:status:RequestDenied`.

Процедуры взаимодействия HTTP в ходе обмена сообщениями не должны использовать коды состояния ошибки HTTP для указания ошибки в обработке SAML, поскольку агент пользователя не является полной стороной в протокольном обмене SAML (см. также раздел 9).

### 10.2.4.7 Аспекты метаданных

Поддержку связи перенаправления HTTP следует обозначить, указав оконечные точки URL в которые должны быть переданы запросы и ответы для конкретного протокола или профиля. Могут быть указаны либо одна оконечная точка, либо различные оконечные точки для запроса и ответа.

ПРИМЕЧАНИЕ (информативное). – PE2 (см. OASIS PE:2006) предлагает заменить вышеприведенный параграф следующим текстом:

Поддержку получения сообщений с использованием связи артефакта HTTP следует обозначить, указав оконечные точки URL в которые должны быть переданы запросы и ответы для конкретного протокола или профиля. Могут быть указаны либо одна оконечная точка, либо различные оконечные точки для запроса и ответа. Поддержка передачи сообщений с использованием этой связи должна сопровождаться одним или несколькими пронумерованными сообщениями `<md:ArtifactResolutionService>` оконечные точки для обработки `<samlp:ArtifactResolve>`.

### 10.2.4.8 Пример обмена сообщениями SAML с использованием перенаправления HTTP

В этом примере – обмен парой сообщений `<LogoutRequest>` и `<LogoutResponse>` с использованием связи перенаправления HTTP.

Сначала приведем реальное передаваемое сообщение SAML:

```
<samlp:LogoutRequest xmlns:samlp="urn:oasis:names:tc:SAML:2.0:protocol"
xmlns="urn:oasis:names:tc:SAML:2.0:assertion"
    ID="d2b7c388cec36fa7c39c28fd298644a8" IssueInstant="2004-01-
21T19:00:49Z" Version="2.0">
    <Issuer>https://IdentityProvider.com/SAML</Issuer>
    <NameID Format="urn:oasis:names:tc:SAML:2.0:nameid-
format:persistent">005a06e0-ad82-110d-a556-004005b13a2b</NameID>
    <samlp:SessionIndex>1</samlp:SessionIndex>
</samlp:LogoutRequest>

<samlp:LogoutResponse xmlns:samlp="urn:oasis:names:tc:SAML:2.0:protocol"
xmlns="urn:oasis:names:tc:SAML:2.0:assertion"
    ID="b0730d21b628110d8b7e004005b13a2b"
InResponseTo="d2b7c388cec36fa7c39c28fd298644a8"
    IssueInstant="2004-01-21T19:00:49Z" Version="2.0">
    <Issuer>https://ServiceProvider.com/SAML</Issuer>
```

```
    <samlp:Status>
        <samlp:StatusCode
Value="urn:oasis:names:tc:SAML:2.0:status:Success"/>
    </samlp:Status>
</samlp:LogoutResponse>
```

Исходный запрос HTTP от агента пользователя в вышеприведенном примере этой связью не определяется. Для инициирования отмены протокольного обмена запрашивающая сторона SAML возвращает следующий ответ HTTP, содержащий подписанное Сообщение-запрос SAML. Значение параметра `SAMLRequest` действительно выводится из вышеприведенного сообщения-запроса. Участок подписи приведен только для иллюстрации и не является результатом реальных вычислений. Переводы строки в заголовке HTTP `Location` ниже – это артефакт документа, а в реальном заголовке нет никаких переводов строки.

```
HTTP/1.1 302 Object Moved
Date: 21 Jan 2004 07:00:49 GMT
Location:
https://ServiceProvider.com/SAML/SLO/Browser?SAMLRequest=fVFdS8MwFH0f7D%2BUv
GdNsq62oSsIQyhMESc%2B%2BJYlmRbWpObeyvz3puv2IMjyFM7HPedyK1DdsZdb%2F%2BEHfLFfg
wVMTt3RgTwzazIEJ72CFqRTnQWJWu7uH7dSLJjsg0ev%2FZFMlttiBWADtt6R%2BSyJr9msiRH7O
70sCm31Mj%2Bo%2BC%2B1KA5GlEWeZaogSQMw2MYBKodrIhjLKONU8FdeSsZkVr6T5M0GiHMjvWC
knqZXZ2OoPxF7kGnaGOuwxZ%2Fn4L9bY8NC%2By4du1XpRXnxPcXizSZ58KFTeHujEWkNPZylsh9
bAMYYUjO2Uiy3jCpTCMo5M1StVjmN9SO150sl9lU6RV2Dp0vsLIy7NM7YU82r9B90PrvCf85W%2F
wL8zSVQzAEAAA%3D%3D&RelayState=0043bfc1bc45110dae17004005b13a2b&SigAlg=http%
3A%2F%2Fwww.w3.org%2F200%2F09%2Fxmldsig%23rsa-
sha1&Signature=NOTAREALSIGNATUREBUTTHEREALONEWOULDGOHERE
Content-Type: text/html; charset=iso-8859-1
```

После любых неопределенных спецификацией взаимодействий, которые могут случиться, отвечающая сторона SAML возвращает показанный ниже ответ HTTP, содержащий подписанное сообщение-ответ SAML. Повторим, значение параметра `SAMLResponse` действительно выводится из вышеприведенного сообщения-запроса. Участок подписи приведен только для иллюстрации и не является результатом реальных вычислений.

```
HTTP/1.1 302 Object Moved
Date: 21 Jan 2004 07:00:49 GMT
Location:
https://IdentityProvider.com/SAML/SLO/Response?SAMLResponse=fVFNa4QwEL0X%2Bh
8k912TaDUGFUp7EbZQ6rKH3mKcbQVNJBOX%2FvxaXQ9tYec0vHlv3nzkqIZ%2BlAf7YSf%2FBjha
gxB8Db1BuZQKMjkjrcIOpVEDoPRa1o8vB8n3VI7OeqttT1bJbbJCBOc7a8j9XTBH9VyQhqYRbTlr
Ei4Yo61oUqA0pvShYZHiDQkqs411tAVpeZPqSAgNOkrOas4zzcW55ZlI4liJrTXiBJVBr4wvCJ87
7ijbcXZkmaRUxtk7CU7gcB5mLu8pKVddvghd%2Ben9iDIMa3CXTsOrs5euBbfXdgh%2F9snDK%2F
EqW69Ye%2BUnvGL%2F8CfbQnBS%2FQS3z4QLW9aT1oBIws0j%2FGOyAb9%2FV34Dw5k779IBAAA%
3D&RelayState=0043bfc1bc45110dae17004005b13a2b&SigAlg=http%3A%2F%2Fwww.w3.or
g%2F200%2F09%2Fxmldsig%23rsa-
sha1&Signature=NOTAREALSIGNATUREBUTTHEREALONEWOULDGOHERE
Content-Type: text/html; charset=iso-8859-1
```

### 10.2.5 Связь HTTP POST

Связь HTTP POST определяет механизм, при помощи которого протокольные сообщения SAML могут передаваться внутри содержания управляющего сообщения HHTP в кодировке base64.

Эта связь может быть образована с использованием связи перенаправления HTTP (см. 10.2.4) и связи артефакта HTTP (см. 10.2.6) для передачи сообщения запроса и ответа в одном-единственном обмене сообщениями с использованием двух различных связей.

#### 10.2.5.1 Требуемая информация

**Идентификация**: `urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST`

**Контактная информация**: security-services-comment@lists.oasis-open.org

**Описание**: Приводится ниже.

**Обновления**: Нет.

#### 10.2.5.2 Обзор

Связь HTTP POST предназначена для случаев, при которых запрашивающая сторона SAML и отвечающая сторона SAML вынуждены связываться друг с другом, используя в качестве промежуточного элемента агент пользователя HTTP (определенный в IETF RFC 2616). Это может потребоваться, например, если связывающиеся стороны не имеют общего прямого пути для связи. Это может потребоваться также, если отвечающая сторона нуждается во взаимодействии с агентом пользователя для того, чтобы выполнять этот запрос, например, когда агент пользователя должен ее аутентифицировать

Некоторые агенты пользователя HTTP могут иметь возможность исполнять более активную роль в протоколе обмена и могут поддерживать другие связи, использующие HTTP, например связи SOAP и обратная SOAP. Эта связь, кроме возможностей обычного веб-браузера, предполагает возможность передачи уведомлений.

#### 10.2.5.3 RelayState

Данные RelayState могут быть включены вместе с протокольным сообщением SAML, передаваемым при помощи этой связи. Их длина не должна превышать 80 байтов, и их целостность должна быть защищена элементом, создающим сообщение, независимое от любых других защит, которые могут существовать или не существовать во время передачи сообщения. Подписание не реализуемо, учитывая ограничения объекта, но, поскольку существует угроза искажения этого значения третьей стороной, этот элемент должен обеспечить невозможность его искажения путем использования проверочной суммы, псевдослучайного значения или аналогичных средств.

Если сообщение-запрос SAML сопровождается данными RelayState, то отвечающая сторона SAML должна возвратить свой протокольный ответ SAML, используя связь, которая также поддерживает механизм RelayState, и она должна поместить точно те же данные RelayState, которые она получила с запросом, в соответствующий параметра RelayState ответа.

Если в сообщение-запрос SAML не включено ни одного такого значения, или если сообщение-ответ SAML создается без соответствующего запроса, то отвечающая сторона SAML может включить данные RelayState, которые должны быть интерпретированы получателем на основе использования профиля или предварительного соглашения между сторонами.

ПРИМЕЧАНИЕ (информативное). – PE31 (см. OASIS PE:2006) предлагает разъяснить вышеприведенный параграф, как показано ниже:

> Если никаких параметров RelayState не включено в сообщение-запрос SAML, или если сообщение-ответ SAML создается без соответствующего запроса, то отвечающая сторона SAML может включить данные RelayState, которые должны быть интерпретированы получателем на основе использования профиля либо на основе предварительного соглашения между сторонами.

#### 10.2.5.4 Кодирование сообщений

Для использования с этой связью сообщения кодируются кодировкой XML в форме управления HTML и передаются с использованием метода HTTP POST. Протокольное сообщение SAML – кодировано при помощи применения правил кодирования base-64 к представлению сообщения в форме XML и помещению результата в скрытой форме контроля внутри самой формы, как определено в W3C HTML (раздел 17). Документ HTML должен придерживаться W3C XHTML в соответствии обычной практикой.

Если сообщение является запросом SAML, то должна быть форма контроля, имеющая название `SAMLRequest`. Если сообщение является ответом SAML, то должна быть форма контроля, имеющая название `SAMLResponse`. Могут быть включены любые другие формы контроля или презентации, но они не должны быть требуемыми, для того чтобы получатель обрабатывал сообщение.

Если значение "RelayState" сопровождает сообщение протокола SAML, то оно должно быть помещено в дополнительный скрытый инструмент управления формы, имеющий название `RelayState` внутри той же формы, что и сообщения SAML.

Атрибутом `action` данной формы должна быть оконечная точка HTTP получателя для использующих эту связь протокола или профиля, на которые должно быть доставлено сообщение SAML. Атрибутом `method` должен быть "POST".

Любой метод, поддерживаемый агентом пользователя, может использоваться для передачи этой формы, и в нее может быть включено содержание любой формы, необходимое для выполнения этого, например, передача параметров управления и команд со стороны клиента. Однако получатель должны иметь возможность обработать сообщение вне зависимости от механизма, при помощи которого была инициирована передача этой формы.

Любые включенные инструмент управления формы должны быть преобразованы так, чтобы их было безопасно вводить в документ XHTML. Это предполагает преобразование таких символов, как кавычки, в элементы HTML и т. д.

#### 10.2.5.5 Обмен сообщениями

Модель системы, используемая для переговоров на языке SAML при помощи этой связи, это модель – запрос-ответ, но эти сообщения передаются агенту пользователя в ответе HTTP и доставляются получателю сообщения в запросе HTTP. Процедуры взаимодействия HTTP до, между и после того, как эти обмены имеют место, в спецификации не определены. Предполагается, что и запрашивающая сторона SAML, и отвечающая сторона SAML являются отвечающими сторонами HTTP. На рисунке 10-3 показаны передаваемые сообщения.

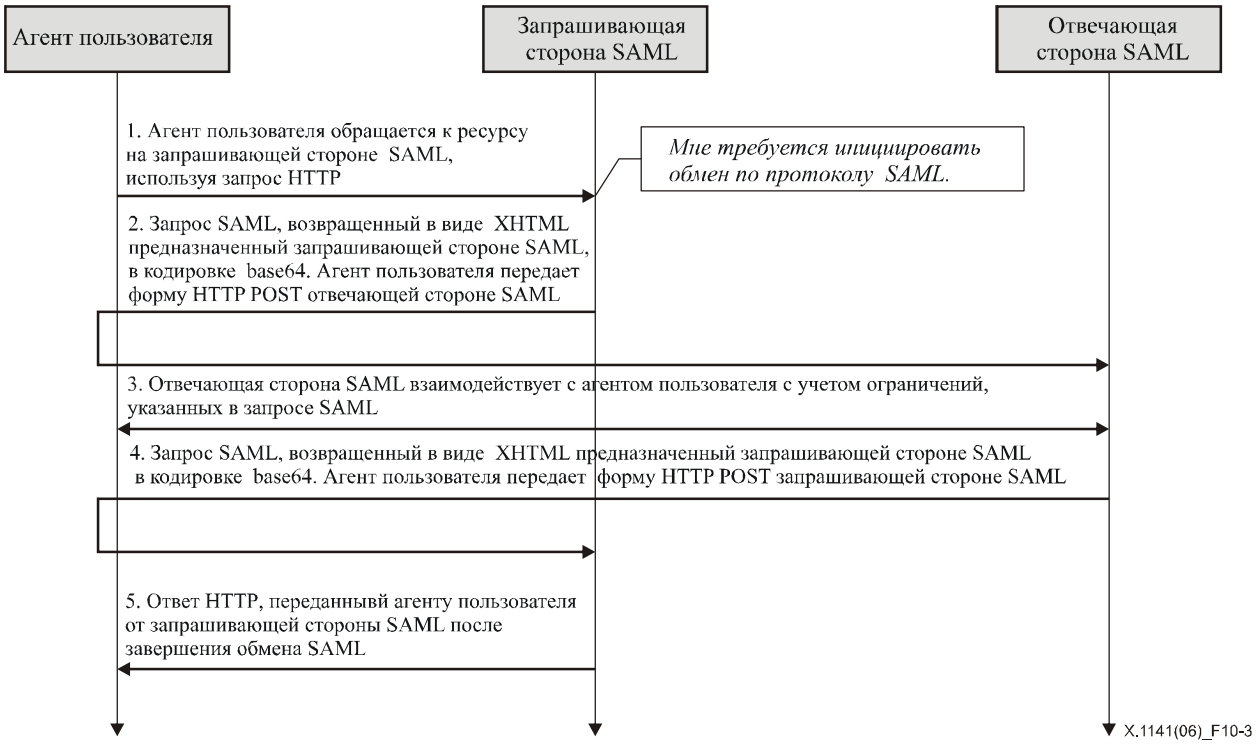


**Рисунок 10-3/X.1141 – Обмен сообщениями HTTP POST**

1) Изначально, агент пользователя передает произвольный запрос HTTP на элемент системы. В процессе обработки запроса, элемент системы решает инициировать обмен по протоколу SAML.
2) Элемент системы, действующий в качестве запрашивающей стороны SAML, отвечает на запрос HTTP от агента пользователя, возвращая запрос SAML. Этот запрос возвращается в виде документа XHTML, содержащего форму и контент, определенные в 10.2.5.4. Агент пользователя доставляет запрос SAML, передавая отвечающей стороне SAML запрос HTTP POST.
3) Как правило, отвечающая сторона SAML может ответить на запрос SAML сразу же, возвращая ответ SAML, или она может вернуть произвольное содержание для упрощения последующего взаимодействия с агентом пользователя, который нужен для выполнения этого запроса. Конкретные протоколы и профили могут содержать механизмы для обозначения уровня желания запрашивающей стороны разрешить такой тип взаимодействия (например, атрибут `IsPassive` в запросе `<samlp:AuthnRequest>`).
4) В итоге отвечающая сторона должна вернуть агенту пользователя ответ SAML, который дожжен быть возвращен запрашивающей стороне SAML. Ответ SAML возвращается точно таким же образом, который описан для запроса SAML на этапе 2.
5) После получения ответа SAML запрашивающая сторона SAML возвращает произвольный ответ HTTP агенту пользователя.

#### 10.2.5.5.1 HTTP и аспекты кэширования

Прокси-элементы HTTP и промежуточные элементы агента пользователя не должны кэшировать протокольные сообщения SAML. Для обеспечения этого, должны выполняться следующие правила.

Возвращая протокольные сообщения SAML с использованием HTTP 1.1, отвечающие стороны HTTP должны:

- включить поле заголовка `Cache-Control`, установленное в значение "`no-cache, no-store`";
- включить поле заголовка `Pragma`, установленное в значение "`no-cache`".

Не существует иных ограничений на использование заголовков HTTP.

#### 10.2.5.5.2 Аспекты безопасности

Существование промежуточного элемента агента пользователя означает, что запрашивающая сторона и отвечающая сторона не могут надеяться на транспортный уровень для обеспечения сквозной аутентификация, целостности и конфиденциальности и должны аутентифицировать принятые сообщения. SAML предусматривает в таких случаях подписание протокольных сообщений для обеспечения аутентификации и целостности. Кодированные в определенной форме сообщения могут быть подписаны до применения кодирования base64.

Если сообщение подписано, то атрибут XML `Destination` в корневом элементе протокольного сообщения SAML должен содержать URL, куда отправитель приказывает агенту пользователя доставить сообщение. Получатель должен затем проверить, чтобы это значение соответствовало местоположению, из которого было получено сообщение.

Эта связь не должна использоваться, если содержание запроса или ответа не должно быть раскрыто промежуточному элементу агента пользователя. В противном случае требование конфиденциальности и запросов SAML, и ответов SAML является дополнительным и зависит от условий использования. Если конфиденциальность необходима, то должен использоваться TLS 1.0 для защиты сообщения при передаче между агентом пользователя и запрашивающей стороной или отвечающей стороной SAML.

Как правило, эта связь опирается на аутентификацию и защиту целостности на уровне сообщения за счет подписания и не поддерживает конфиденциальность сообщений от промежуточного элемента агента пользователя.

Не определено механизма для защиты целостности взаимосвязи между сообщением протокола SAML и значением "RelayState", если оно имеется. То есть атакующая сторона может перекомбинировать пару достоверных ответов протокола HTTP, переключая значения "RelayState", ассоциированные с каждым протокольным сообщением SAML. Может быть обеспечена целостность отдельно значения "RelayState" и сообщения SAML, но не их комбинации. В результате, создатель и потребитель информации "RelayState" должен проявлять осторожность и не связывать чувствительную информацию о состоянии со значением "RelayState" без принятия соответствующих мер предосторожности (на основе информации в сообщении SAML).

#### 10.2.5.6 Сообщение об ошибках

Отвечающая сторона SAML, которая отказывается выполнять обмен сообщениями с запрашивающей стороной SAML, должна вернуть сообщение-ответ SAML со значением кода второго уровня `<samlp:StatusCode>` = `urn:oasis:names:tc:SAML:2.0:status:RequestDenied`.

Процедуры взаимодействия HTTP в ходе обмена сообщениями не должны использовать коды состояния ошибки HTTP для указания ошибки в обработке SAML, поскольку агент пользователя не является полной стороной в протокольном обмене SAML.

Более подробная информация о кодах состояния SAML приведена в разделе 8.2.

#### 10.2.5.7 Аспекты метаданных

Поддержку связи HTTP POST следует обозначить, указав оконечные точки URL в которые должны быть переданы запросы и ответы для конкретного протокола или профиля. Могут быть указаны либо одна оконечная точка, либо различные оконечные точки для запроса и ответа.

#### 10.2.5.8 Пример обмена сообщениями SAML с использованием HTTP POST

В этом примере – обмен парой сообщений `<LogoutRequest>` и `<LogoutResponse>` с использованием связи HTTP POST.

Сначала приведем реальное передаваемое протокольные сообщения SAML:

```
<samlp:LogoutRequest xmlns:samlp="urn:oasis:names:tc:SAML:2.0:protocol"
xmlns="urn:oasis:names:tc:SAML:2.0:assertion"
    ID="d2b7c388cec36fa7c39c28fd298644a8" IssueInstant="2004-01-
21T19:00:49Z" Version="2.0">
    <Issuer>https://IdentityProvider.com/SAML</Issuer>
    <NameID Format="urn:oasis:names:tc:SAML:2.0:nameid-
format:persistent">005a06e0-ad82-110d-a556-004005b13a2b</NameID>
    <samlp:SessionIndex>1</samlp:SessionIndex>
</samlp:LogoutRequest>

<samlp:LogoutResponse xmlns:samlp="urn:oasis:names:tc:SAML:2.0:protocol"
xmlns="urn:oasis:names:tc:SAML:2.0:assertion"
    ID="b0730d21b628110d8b7e004005b13a2b"
InResponseTo="d2b7c388cec36fa7c39c28fd298644a8"
    IssueInstant="2004-01-21T19:00:49Z" Version="2.0">
    <Issuer>https://ServiceProvider.com/SAML</Issuer>
    <samlp:Status>
        <samlp:StatusCode
Value="urn:oasis:names:tc:SAML:2.0:status:Success"/>
    </samlp:Status>
</samlp:LogoutResponse>
```

Исходный запрос HTTP от агента пользователя в вышеприведенном примере этой связью не определяется. Для инициирования отмены протокольного обмена запрашивающая сторона SAML возвращает следующий ответ HTTP, содержащий подписанное Сообщение-запрос SAML. Значение параметра `SAMLRequest` действительно выводится из вышеприведенного сообщения-запроса.

```
HTTP/1.1 200 OK
Date: 21 Jan 2004 07:00:49 GMT
Content-Type: text/html; charset=iso-8859-1

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN"
"http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en">
<body onload="document.forms[0].submit()">

<noscript>
<p>
<strong>Note:</strong> Since your browser does not support JavaScript, you
must press the Continue button once to proceed.
</p>
</noscript>

<form action="https://ServiceProvider.com/SAML/SLO/Browser" method="post">
<div>
<input type="hidden" name="RelayState"
value="0043bfc1bc45110dae17004005b13a2b"/>
<input type="hidden" name="SAMLRequest"
value="PHNhbWxwOkxvZ291dFJlcXVlc3QgeG1sbnM6c2FtbHA9InVybjpvYXNpczpuYW1l
czp0YzpTQU1MOjIuMDpwcm90b2NvbCIgeG1sbnM9InVybjpvYXNpczpuYW1lczp0
YzpTQU1MOjIuMDphc3NlcnRpb24iDQogICAgSUQ9ImQyYjdjMzg4Y2VjMzZmYTdj
MzljMjhmZDI5ODY0NGE4IiBJc3N1ZUluc3RhbnQ9IjIwMDQtMDEtMjFUMTk6MDA6
NDlaIiBWZXJzaW9uPSIyLjAiPg0KICAgIDxJc3N1ZXI+aHR0cHM6Ly9JZGVudGl0
eVByb3ZpZGVyLmNvbS9TQU1MPC9Jc3N1ZXI+DQogICAgPE5hbWVJRCBGb3JtYXQ9
InVybjpvYXNpczpuYW1lczp0YzpTQU1MOjIuMDpuYW1laWQtZm9ybWF0OnBlcnNp
c3RlbnQiPjAwNWEwNmUwLWFkODItMTEwZC1hNTU2LTAwNDAwNWIxM2EyYjwvTmFt
ZUlEPg0KICAgIDxzYW1scDpNb2dvdXRSZXF1ZXN0Pg==" />
</div>
<noscript>
<div>
<input type="submit" value="Continue"/>
</div>
</noscript>
</form>
</body>
</html>
```

После любых неопределенных спецификацией взаимодействий, которые могут случиться, отвечающая сторона SAML возвращает показанный ниже ответ HTTP, содержащий подписанное сообщение-ответ SAML. Повторим, значение параметра `SAMLResponse` действительно выводится из вышеприведенного сообщения-запроса.

```
HTTP/1.1 200 OK
Date: 21 Jan 2004 07:00:49 GMT
Content-Type: text/html; charset=iso-8859-1

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN"
"http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en">
<body onload="document.forms[0].submit()">

<noscript>
<p>
<strong>Note:</strong> Since your browser does not support JavaScript, you
must press the Continue button once to proceed.
</p>
</noscript>

<form action="https://IdentityProvider.com/SAML/SLO/Response" method="post">
<div>
<input type="hidden" name="RelayState"
value="0043bfc1bc45110dae17004005b13a2b"/>
<input type="hidden" name="SAMLResponse"
value="PHNhbWxwOkxvZ291dFJlc3BvbnNlIHhtbG5zOnNhbWxwPSJ1cm46b2FzaXM6bmFt
```

```
ZXM6dGM6U0FNTDoyLjA6cHJvdG9jb2wiIHhtbG5zPSJ1cm46b2FzaXM6bmFtZXM6
dGM6U0FNTDoyLjA6YXNzZXJ0aW9uIg0KICAgIElEPSJiMDczMGQyMWI2MjgxMTBk
OGI3ZTAwNDAwNWIxM2EyYiIgSW5SZXNwb25zZVRvPSJkMmI3YzM4OGNlYzM2ZmE3
YzM5YzI4ZmQyOTg2NDRhOCINCiAgICBJc3N1ZUluc3RhbnQ9IjIwMDQtMDEtMjFU
MTk6MDA6NDlaIiBWZXJzaW9uPSIyLjAiPg0KICAgIDxJc3N1ZXI+aHR0cHM6Ly9T
ZXJ2aWNlUHJvdmlkZXIuY29tL1NBTUw8L0lzc3Vlcj4NCiAgICA8c2FtbHA6U3Rh
dHVzPg0KICAgICAgICA8c2FtbHA6U3RhdHVzQ29kZSBWYWx1ZT0idXJuOm9hc2lz
Om5hbWVzOnRjOlNBTUw6Mi4wOnN0YXR1czpTdWNjZXNzIi8+DQogICAgPC9zYW1s
cDpTdGF0dXM+DQo8L3NhbWxwOlN0YXR1c3BvbnNlPg==" />
</div>
<noscript>
<div>
<input type="submit" value="Continue"/>
</div>
</noscript>
</form>
</body>
</html>
```

### 10.2.6 Связь HTTP Artifact

В связи HTTP Artifact запрос SAML, ответ SAML, или она оба передаются посредством ссылок, с использованием маленького события, называемого артефактом. Отдельная синхронная связь, такая как связь SAML SOAP, используется для обмена артефакта на реальное сообщение протокола с использованием протокола разрешения артефакта, определенного в разделе 8.

Эта связь может быть образована с использованием связи перенаправления HTTP (см. 10.2.4) и связи HTTP POST (см. 10.2.5) для передачи сообщения запроса и ответа в одном-единственном обмене сообщениями с использованием двух различных связей.

#### 10.2.6.1 Требуемая информация

**Идентификация**: `urn:oasis:names:tc:SAML:2.0:bindings:HTTP-Artifact`

**Контактная информация**: security-services-comment@lists.oasis-open.org

**Описание**: Приводится ниже.

**Обновления**: Нет.

#### 10.2.6.2 Обзор

Связь HTTP предназначена для случаев, при которых запрашивающая сторона SAML и отвечающая сторона SAML вынуждены связываться друг с другом, используя в качестве промежуточного элемента агент пользователя HTTP, но ограничения промежуточного элемента препятствуют или противодействуют передаче через него полного сообщения (или обмену сообщениями). Тому могут быть технические причины или нежелание открывать промежуточному элементу содержание сообщения (и если невозможно применить шифрование).

Из-за необходимости впоследствии распознавать этот артефакт, используя другую синхронную связь, например SOAP, должна существовать прямая связь между отправителем и получателем сообщения SAML в обратном направлении передачи артефакта (приемник сообщения и артефакта должен иметь возможность передать запрос `<samlp:ArtifactResolve>` обратно создателю артефакта). Создатель артефакта также должен поддерживать состояние, пока артефакт рассматривается, что имеет свои последствия в условиях выравнивание нагрузки.

#### 10.2.6.3 Кодирование сообщений

Существует два метода кодирования артефакта для использования с этой связью. Один заключается в том, что артефакт кодируется в параметр URL, а второй – в том, что артефакт помещается в форму управления HTML. Когда используется кодирование URL, для доставки сообщения применяется метод HTTP GET, а метод POST используется с кодированием формы. Все оконечные точки, которые поддерживают эту связь, должны поддерживать оба метода.

##### 10.2.6.3.1 RelayState

Данные RelayState могут быть включены в артефакт SAML, передаваемый при помощи этой связи. Их длина не должна превышать 80 байтов, и их целостность должна быть защищена элементом, создающим сообщение, независимое от любых других защит, которые могут существовать или не существовать во время передачи сообщения. Подписание не реализуемо, учитывая ограничения объекта, но, поскольку существует угроза искажения этого значения третьей стороной, этот элемент должен обеспечить невозможность его искажения путем использования проверочной суммы, псевдослучайного значения или аналогичных средств.

Если артефакт, который представляет запрос SAML, сопровождается данными RelayState, то отвечающая сторона SAML должна возвратить свой протокольный ответ SAML, используя связь, которая также поддерживает механизм RelayState, и она должна поместить точно те же данные RelayState, которые она получила с артефактом, в соответствующий параметр RelayState ответа.

Если ни одного такого значения не включено в артефакт, представляющий собой запрос SAML, или если сообщение-ответ SAML создается без соответствующего запроса, то отвечающая сторона SAML может включить данные RelayState, которые должны быть интерпретированы получателем на основе использования профиля или предварительного соглашения между сторонами.

#### 10.2.6.3.2 Кодирование URL

Для кодирования артефакта в URL значение артефакта переводится в кодировку URL и помещается в параметр строки запроса, имеющий название `SAMLart`.

Если значение "RelayState" должно сопровождать артефакт SAML, оно должно быть в кодировке URL и размещаться в дополнительном параметре строки запроса, имеющем название `RelayState`.

#### 10.2.6.3.3 Кодирование в виде формы

Артефакт SAML кодируется в виде формы путем помещения его в скрытый инструмент управления формы, определенной в W3C HTML. Документ HTML должен соответствовать W3C XHTML. Этот инструмент управления формы должен иметь название `SAMLart`. Могут быть включены любые дополнительные инструменты управления формы или представления, но они не должны быть необходимыми для того, чтобы получатель мог обработать артефакт.

Если значение "RelayState" должно сопровождать артефакт SAML, то оно должно быть помещено в дополнительный скрытый инструмент управления формы, имеющий название `RelayState` внутри той же формы, что и сообщения SAML.

Атрибутом `action` данной формы должна быть оконечная точка HTTP получателя для использующих эту связь протокола или профиля, на которые должен быть доставлен этот артефакт. Атрибутом `method` должен быть "POST".

Любой метод, поддерживаемый агентом пользователя, может использоваться для передачи этой формы, и в нее может быть включено содержание любой формы, необходимое для выполнения этого, например, передача параметров управления и команд со стороны клиента. Однако получатель должны иметь возможность обработать сообщение вне зависимости от механизма, при помощи которого была инициирована передача этой формы.

Любые включенные инструмент управления формы должны быть преобразованы так, чтобы их было безопасно вводить в документ XHTML. Это предполагает преобразование таких символов, как кавычки, в элементы HTML и т. д.

### 10.2.6.4 Формат артефакта

Для этой связи артефакт – это короткая скрытая строка. Могут быть определены и могут использоваться различные типы артефактов, не оказывая влияния на связь. Существенными характеристиками является способность приемника артефакта определить создателя артефакта, противостоять взлому и подделкам, его уникальность и компактность.

Общий формат любого артефакта включает в себя обязательный двухбайтовый код типа артефакта и двухбайтовый индекс, идентифицирующий конкретную оконечную точку получения данных о создателе артефакта, следующего вида:

```
SAML_artifact     := B64(TypeCode EndpointIndex RemainingArtifact)
TypeCode          := Byte1Byte2
EndpointIndex     := Byte1Byte2
```

Обозначение `B64(TypeCode EndpointIndex RemainingArtifact)` обозначает преобразование приложения base64 (см. IETF RFC 2045) в цепочку типа `Code`, `EndpointIndex`, и `RemainingArtifact`.

Для создания артефакта SAML рекомендуются следующие действия.

- Каждому создателю назначается идентифицирующий URI, известный также как идентификатор (ID) создателя (или провайдера). Этот тип идентификатора рассмотрен в разделе 8.
- Создатель создает компонент артефакта `SourceID`, взяв контрольную сумму SHA-1 из идентификатора URL. Величина контрольной суммы в шестнадцатеричном виде не кодируется.

  ПРИМЕЧАНИЕ 1. – NIST (Национальный институт стандартов и технологии) сегодня приветствует использование SHA-256 (защищенный алгоритм хеширования с закодированными 256-битовыми ключами) вместо SHA-1.
- Значение сообщения `Handle` создается из криптостойких случайных и псевдослучайных цифровых последовательностей (см. IETF RFC 1750), генерируемых создателем. Эта последовательность состоит из величин размером как минимум 16 байтов. Эти значения, при необходимости, должны быть заполнены до полной длины 20 байтов.

  ПРИМЕЧАНИЕ 2 (информативное). – PE4 (см. [OASIS Errata Document]) предлагает добавить следующий текст в конец вышеприведенного параграфа:

Хотя общая структура артефакта напоминает те, что использовались в предыдущих версиях SAML, а тип кода одного описанного ниже формата не конфликтует с ранее определенными форматами, не существует явного соответствия между артефактами SAML 2.0 и артефактами любой предыдущей спецификации, и с этой связью не должны использоваться форматы артефактов, которые не были определены специально для использования с SAML 2.0.

Далее описывается один тип артефакта, определенный в SAML V2.0.

#### 10.2.6.4.1 Требуемая информация

**Идентификация**: `urn:oasis:names:tc:SAML:2.0:artifact-04`

**Контактная информация**: security-ervices-comment@lists.oasis-open.org

**Описание**: Приводится ниже.

**Обновления**: Нет.

#### 10.2.6.4.2 Подробное описание формата

SAML V2.0 определяет тип артефакта как тип кода 0x0004. Этот тип артефакта определяется следующим образом:

```
TypeCode          := 0x0004
RemainingArtifact := SourceID MessageHandle
SourceID          := 20-byte_sequence
MessageHandle     := 20-byte_sequence
```

`SourceID` – это 20-байтовая последовательность, используемая приемников артефакта для определения идентификации создателя артефакта и установки множества возможных оконечных точек приема.

Предполагается, что на сайте получателя будет находиться таблица значений `SourceID`, а также одна или несколько оконечных точек с индексами URL (или адресов) для соответствующей отвечающей стороны SAML. Для этой цели может использоваться раздел 9. При приеме артефакта SAML, приемник определяет, принадлежит ли `SourceID` известному создателю артефакта, и получает данные о месте нахождения отвечающей стороны SAML, используя `EndpointIndex`, после чего отправляет по этому адресу сообщение SAML `<samlp:ArtifactResolve>`.

Для любых создателя артефактов, использующие общие приемник, должны использовать различные значения `SourceID`. При формировании значения `MessageHandle` учитывается тот принцип, что они должны иметь предсказываемую взаимосвязь с содержанием указываемого сообщения на стороне его создания и не должны давать возможности восстановить или угадать значение сообщения.

### 10.2.6.5 Обмен сообщениями

Модель системы, используемая для переговоров на языке SAML при помощи этой связи, это модель – запрос-ответ, в которой ссылка на артефакт замещает содержание реального сообщения, и ссылка на артефакт передается агенту пользователя в ответе HTTP и доставляются получателю сообщения в запросе HTTP. Процедуры взаимодействия HTTP до, между и после того, как эти обмены имеют место, в спецификации не определены. Предполагается, что и запрашивающая сторона SAML, и отвечающая сторона SAML являются отвечающими сторонами HTTP.

Кроме того, предполагается, что при получении артефакта через агента пользователя, получатель начинает отдельный прямой обмен сообщениями с создателем артефакта, используя протокол Resolution, определенный в настоящей Рекомендации. Этот обмен должен использовать связь, которая не использует агента пользователя HTTP в качестве промежуточного элемента, такого как связь SOAP. После успешного приема протокольного сообщения SAML, артефакт отбрасывается и продолжается (или завершается, если сообщение является ответом) обработка первоначального обмена по протоколу SAML.

Создание и доставка артефакта вместе с соответствующим шагом разрешения, представляет собой половину всего обмена. Эта связь может использоваться для доставки любой половины, либо всего объема протокольных сообщений SAML. Связь, которая может быть объединена с ней, например перенаправление HTTP (HTTP Redirect) (см. 10.2.4) или связь POST (см. 10.2.5), может использоваться для передачи второй половины обмена сообщениями. Следующая последовательность предполагает, что связь артефакта используется для обеих половин. Рисунок 10-4 ниже иллюстрирует передаваемые сообщения.

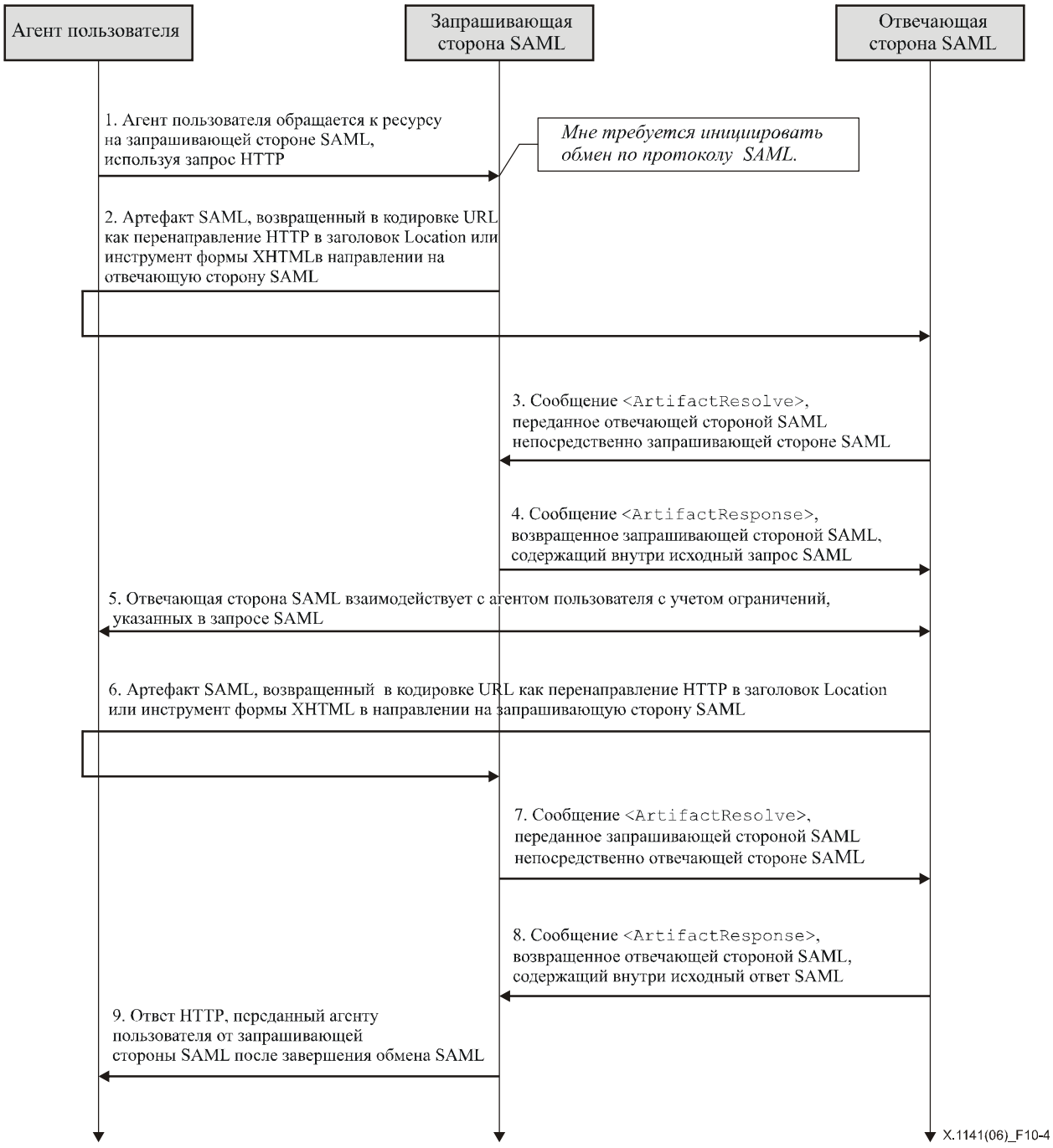


**Рисунок 10-4/X.1141 – Обмен сообщениями артефакта HTTP**

1) Изначально агент пользователя передает произвольный запрос HTTP на элемент системы. В процессе обработки запроса, элемент системы решает инициировать обмен по протоколу SAML.

2) Элемент системы, действующий в качестве запрашивающей стороны SAML, отвечает на запрос HTTP от агента пользователя, возвращая артефакт, представляющий собой запрос SAML.

   - Если артефакт в кодировке URL возвращается закодированным в заголовок Location ответа HTTP и статус HTTP должен быть либо 303, либо 302. Запрашивающая сторона SAML может включить в ответ HTTP дополнительную презентацию и содержание для упрощения задачи агента пользователя по передаче сообщения, как определено в IETF RFC 2616. Агент пользователя доставляет артефакт, отправляя отвечающей стороне SAML запрос HTTP GET.
   - Если он закодирован в виде формы, то артефакт возвращается в документе XHTML, содержащем форму и содержание, определенные в 10.2.6.3.3. Агент пользователя доставляет артефакт, отправляя отвечающей стороне SAML запрос HTTP POST.

3) Отвечающая сторона SAML определяет запрашивающую сторону SAML, изучая артефакт (точный процесс зависит от типа артефакта), и передает запрашивающей стороне SAML запрос `<samlp:ArtifactResolve>`, содержащий артефакт для использования прямой связи SAML, временно меняя роли местами.

4) Предполагая, что все необходимые условия выполняются, запрашивающая сторона SAML возвращает ответ `<samlp:ArtifactResponse>`, содержащий исходное сообщение-запрос SAML, которое должна обработать отвечающая сторона SAML.

5) Как правило, отвечающая сторона SAML может ответить на запрос SAML, немедленно возвратив артефакт SAML, или может вернуть произвольное содержание для упрощения последующего взаимодействия с агентом пользователя, необходимое для выполнения этого запроса. Конкретные протоколы и профили могут содержать механизмы для указания степени желания запрашивающей стороны разрешить этот вид взаимодействия (например, атрибут `IsPassive` в запросе `<samlp:AuthnRequest>`).

6) В итоге отвечающая сторона должна вернуть агенту пользователя артефакт SAML, который должен быть возвращен запрашивающей стороне SAML. Артефакт ответ SAML возвращается тем же путем, который был описан для артефакта – запроса SAML на этапе 2.

7) Запрашивающая сторона SAML определяет запрашивающую сторону SAML, изучая артефакт, и передает отвечающей стороне SAML запрос `<samlp:ArtifactResolve>`, содержащий артефакт для использования прямой связи SAML, как на этапе 3.

   ПРИМЕЧАНИЕ (информативное). – PE31 (см. OASIS PE:2006) предлагает заменить последнее предложение этапа 6 следующим текстом:

   Запрашивающая сторона SAML определяет запрашивающую сторону SAML, изучая артефакт, и передает отвечающей стороне SAML запрос `<samlp:ArtifactResolve>`, содержащий артефакт, используя синхронную связь SAML, как на этапе 3.

8) Предполагая, что все необходимые условия выполняются, отвечающая сторона SAML возвращает ответ `<samlp:ArtifactResponse>`, содержащий сообщение-ответ SAML. который должна обработать запрашивающая сторона SAML, как на этапе 4.

9) После получения ответа SAML, запрашивающая сторона SAML возвращает агенту пользователя произвольный ответ HTTP.

#### 10.2.6.5.1 HTTP и аспекты кэширования

Прокси–элементы HTTP и промежуточные элементы агента пользователя не должны кэшировать артефакты SAML. Для обеспечения этого, должны выполняться следующие правила.

Возвращая артефакты SAML с использованием HTTP 1.1, отвечающие стороны HTTP должны:

- включить поле заголовка `Cache-Control`, установленное в значение "`no-cache, no-store`";
- включить поле заголовка `Pragma`, установленное в значение "`no-cache`".

Не существует иных ограничений на использование заголовков HTTP.

#### 10.2.6.5.2 Аспекты безопасности

Эта связь использует комбинацию непрямые ссылки на передачу сообщений, за которыми следует прямой обмен для возврата реального сообщения. В результате, не требуется обеспечивать аутентификацию или защиту целостности самой ссылки на сообщение (артефакта), но передача запроса обратного вызова/ответа, который возвращает реальное сообщение, может быть взаимно аутентифицирована, и его целостность должна быть защищена, в зависимости от используемой среды передачи.

Если реальное протокольное сообщение SAML предназначено конкретному получателю, то создатель артефакта должен аутентифицировать отправителя последующего сообщения `<samlp:ArtifactResolve>` до возвращения реального сообщения.

Передача артефакта от агента пользователя и к нему должна быть защищена с обеспечением конфиденциальности; либо должен использоваться TLS 1.0. Передача запроса обратного вызова/ответа, который возвращает реальное сообщение, может быть защищена, в зависимости от используемой среды передачи.

Как правило, эта связь опирается на артефакт как на кратковременную ссылку, которую трудно подделать и применяет другие меры обеспечения безопасности для запроса обратного вызова/ответа, который возвращает реальное сообщение. Все артефакты должны иметь семантику разового использования, реализованную создателем артефакта.

Кроме того, рекомендуется, чтобы приемники артефакта также реализовывали семантику разового использования для значений артефактов, которые они принимают, в целях предотвращения возможности злоумышленников вмешаться в распознавание артефакта при помощи агента пользователя и последующей подачи его на приемник артефакта. Если попытка распознать артефакт не удалась, этот артефакт должен быть помещен в список заблокированных артефактов на период времени, который превышает период принятия, в течение которого создатель артефакта принимает решение относительно этого артефакта.

Не определено механизма для защиты целостности взаимосвязи между артефактом и значением "RelayState", если оно имеется. То есть атакующая сторона может перекомбинировать пару достоверных ответов протокола HTTP, переключая значения "RelayState", ассоциированные с каждым артефактом. В результате, создатель и потребитель информации "RelayState" должен проявлять осторожность и не связывать чувствительную информацию о состоянии со значением "RelayState" без принятия соответствующих мер предосторожности (на основе информации в сообщении протокола SAML, полученного при помощи артефакта).

#### 10.2.6.6 Сообщение об ошибках

Отвечающая сторона SAML , которая отказывается выполнять обмен сообщениями с запрашивающей стороной SAML, должна вернуть сообщение-ответ со значением кода второго уровня `<samlp:StatusCode>` = `urn:oasis:names:tc:SAML:2.0:status:RequestDenied`.

Процедуры взаимодействия HTTP в ходе обмена сообщениями не должны использовать коды состояния ошибки HTTP для указания ошибки в обработке SAML, поскольку агент пользователя не является полной стороной в протокольном обмене SAML.

Если создатель артефакта принимает сообщение `<samlp:ArtifactResolve>`, которое он может понять, он должен возвратить ответ `<samlp:ArtifactResponse>` со значением `<samlp:StatusCode>` = `urn:oasis:names:tc:SAML:2.0:status:Success`, даже если он не возвращает соответствующее сообщение (например, из-за того, что сторона, запрашивающая артефакт, не имеет права получить сообщение, или истек срок действия артефакта).

#### 10.2.6.7 Аспекты метаданных

Поддержку связи HTTP Artifact следует обозначить, указав оконечные точки URL, в которые должны быть переданы запросы и ответы для конкретного протокола или профиля. Могут быть указаны либо одна оконечная точка, либо различные оконечные точки для запроса и ответа. Также должны быть описаны одна или несколько пронумерованных оконечных точек для обработки сообщения `<samlp:ArtifactResolve>`.

#### 10.2.6.8 Пример обмена сообщениями SAML с использованием HTTP Artifacta

В этом примере – обмен парой сообщений `<LogoutRequest>` и `<LogoutResponse>` с использованием связи HTTP Artifact, когда прием артефакты выполняется с использованием связи SOAP, объединенной с HTTP.

Сначала приведем реальное передаваемое протокольные сообщения SAML:

```
<samlp:LogoutRequest xmlns:samlp="urn:oasis:names:tc:SAML:2.0:protocol"
xmlns="urn:oasis:names:tc:SAML:2.0:assertion"
    ID="d2b7c388cec36fa7c39c28fd298644a8" IssueInstant="2004-01-
21T19:00:49Z" Version="2.0">
    <Issuer>https://IdentityProvider.com/SAML</Issuer>
    <NameID Format="urn:oasis:names:tc:SAML:2.0:nameid-
format:persistent">005a06e0-ad82-110d-a556-004005b13a2b</NameID>
    <samlp:SessionIndex>1</samlp:SessionIndex>
</samlp:LogoutRequest>

<samlp:LogoutResponse xmlns:samlp="urn:oasis:names:tc:SAML:2.0:protocol"
xmlns="urn:oasis:names:tc:SAML:2.0:assertion"
    ID="b0730d21b628110d8b7e004005b13a2b"
InResponseTo="d2b7c388cec36fa7c39c28fd298644a8"
    IssueInstant="2004-01-21T19:00:49Z" Version="2.0">
    <Issuer>https://ServiceProvider.com/SAML</Issuer>
    <samlp:Status>
        <samlp:StatusCode
Value="urn:oasis:names:tc:SAML:2.0:status:Success"/>
    </samlp:Status>
</samlp:LogoutResponse>
```

Исходный запрос HTTP от агента пользователя в вышеприведенном примере этой связью не определяется. Для инициирования отмены протокольного обмена запрашивающая сторона SAML возвращает следующий ответ HTTP, содержащий артефакт SAML. Символы перевода строки в показанном ниже заголовке HTTP Location, являются результатом форматирования документа, а в реальном заголовке нет ни одного символа перевода строки.

```
HTTP/1.1 302 Object Moved
Date: 21 Jan 2004 07:00:49 GMT
Location:
https://ServiceProvider.com/SAML/SLO/Browser?SAMLart=AAQAADWNEw5VT47wcO4zX%2
FiEzMmFQvGknDfws2ZtqSGdkNSbsW1cmVR0bzU%3D&RelayState=0043bfc1bc45110dae17004
005b13a2b
Content-Type: text/html; charset=iso-8859-1
```

Затем отвечающая сторона SAML распознает артефакт, который она получает, его в реальный запрос SAML, используя протокол распознавания артефакта и связь SOAP на этапах 3 и 4, следующим образом:

Этап 3:

```
POST /SAML/Artifact/Resolve HTTP/1.1
Host: IdentityProvider.com
Content-Type: text/xml
Content-Length: nnn
SOAPAction: http://www.oasis-open.org/committees/security
<SOAP-ENV:Envelope
    xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
    <SOAP-ENV:Body>
        <samlp:ArtifactResolve
    xmlns:samlp="urn:oasis:names:tc:SAML:2.0:protocol"
    xmlns="urn:oasis:names:tc:SAML:2.0:assertion"
    ID="_6c3a4f8b9c2d" Version="2.0"
    IssueInstant="2004-01-21T19:00:49Z">
    <Issuer>https://ServiceProvider.com/SAML</Issuer>
    <Artifact>
    AAQAADWNEw5VT47wcO4zX/iEzMmFQvGknDfws2ZtqSGdkNSbsW1cmVR0bzU=
    </Artifact>
        </samlp:ArtifactResolve>
    </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Этап 4:

```
HTTP/1.1 200 OK
Date: 21 Jan 2004 07:00:49 GMT
Content-Type: text/xml
Content-Length: nnnn

<SOAP-ENV:Envelope
    xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
    <SOAP-ENV:Body>
        <samlp:ArtifactResponse
    xmlns:samlp="urn:oasis:names:tc:SAML:2.0:protocol"
    xmlns="urn:oasis:names:tc:SAML:2.0:assertion"
    ID="_FQvGknDfws2Z" Version="2.0"
    InResponseTo="_6c3a4f8b9c2d"
    IssueInstant="2004-01-21T19:00:49Z">
    <Issuer>https://IdentityProvider.com/SAML</Issuer>
         <samlp:Status>
       <samlp:StatusCode
    Value="urn:oasis:names:tc:SAML:2.0:status:Success"/>
    </samlp:Status>
    <samlp:LogoutRequest ID="d2b7c388cec36fa7c39c28fd298644a8"
       IssueInstant="2004-01-21T19:00:49Z"
       Version="2.0">
            <Issuer>https://IdentityProvider.com/SAML</Issuer>
            <NameID Format="urn:oasis:names:tc:SAML:2.0:nameid-
format:persistent">005a06e0-ad82-110d-a556-004005b13a2b</NameID>
            <samlp:SessionIndex>1</samlp:SessionIndex>
    </samlp:LogoutRequest>
        </samlp:ArtifactResponse>
    </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

После любых неопределенных спецификацией взаимодействий, которые могут случиться, отвечающая сторона SAML возвращает второй артефакт SAML в своем ответе HTTP на этапе 6:

```
HTTP/1.1 302 Object Moved
Date: 21 Jan 2004 07:05:49 GMT
Location:
https://IdentityProvider.com/SAML/SLO/Response?SAMLart=AAQAAFGIZXv5%2BQaBaE5
qYurHWJO1nAgLAsqfnyiDHIggbFU0mlSGFTyQiPc%3D&RelayState=0043bfc1bc45110dae170
04005b13a2b
Content-Type: text/html; charset=iso-8859-1
```

Затем отвечающая сторона SAML распознает полученный артефакт, превращая его в реальный запрос SAML с использованием протокола распознавания артефакта и связи SOAP на этапах 7 и 8, следующим образом:

Этап 7:

```
POST /SAML/Artifact/Resolve HTTP/1.1
Host: ServiceProvider.com
Content-Type: text/xml
Content-Length: nnn
SOAPAction: http://www.oasis-open.org/committees/security
<SOAP-ENV:Envelope
    xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
    <SOAP-ENV:Body>
        <samlp:ArtifactResolve
    xmlns:samlp="urn:oasis:names:tc:SAML:2.0:protocol"
    xmlns="urn:oasis:names:tc:SAML:2.0:assertion"
    ID="_ec36fa7c39" Version="2.0"
    IssueInstant="2004-01-21T19:05:49Z">
    <Issuer>https://IdentityProvider.com/SAML</Issuer>
    <Artifact>
    AAQAAFGIZXv5+QaBaE5qYurHWJO1nAgLAsqfnyiDHIggbFU0mlSGFTyQiPc=
    </Artifact>
        </samlp:ArtifactResolve>
    </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Этап 8:

```
HTTP/1.1 200 OK
Date: 21 Jan 2004 07:05:49 GMT
Content-Type: text/xml
Content-Length: nnnn

<SOAP-ENV:Envelope
    xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
    <SOAP-ENV:Body>
        <samlp:ArtifactResponse
    xmlns:samlp="urn:oasis:names:tc:SAML:2.0:protocol"
    xmlns="urn:oasis:names:tc:SAML:2.0:assertion"
    ID="_FQvGknDfws2Z" Version="2.0"
    InResponseTo="_ec36fa7c39"
    IssueInstant="2004-01-21T19:05:49Z">
    <Issuer>https://ServiceProvider.com/SAML</Issuer>
         <samlp:Status>
       <samlp:StatusCode
    Value="urn:oasis:names:tc:SAML:2.0:status:Success"/>
    </samlp:Status>
    <samlp:LogoutResponse ID="_b0730d21b628110d8b7e004005b13a2b"
       InResponseTo="_d2b7c388cec36fa7c39c28fd298644a8"
       IssueInstant="2004-01-21T19:05:49Z"
       Version="2.0">
            <Issuer>https://ServiceProvider.com/SAML</Issuer>
       <samlp:Status>
          <samlp:StatusCode
    Value="urn:oasis:names:tc:SAML:2.0:status:Success"/>
       </samlp:Status>
    </samlp:LogoutResponse>
        </samlp:ArtifactResponse>
    </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

### 10.2.7 Связь SAML URI

Идентификаторы URI – это независимые от протокола средства обозначения ресурсов. Эта связь – не обычная связь типа запрос/ответ SAML, она поддерживает инкапсуляцию сообщения `<samlp:AssertionIDRequest>` с одним `<saml:AssertionIDRef>` в ходе разрешения URI (превращения ссылки URI в абсолютную форму URI). Результатом успешного выполнения запроса является элемент SAML `<saml:Assertion>` (а не полный ответ SAML).

Подобно SOAP, разрешение URI может выполняться для многих нижележащих средств транспорта. Эта связь имеет аспекты, не зависящие от транспорта, но также предусматривает использование HTTP с TLS 1.0, как требуется (обязательно для реализации).

ПРИМЕЧАНИЕ (информативное). – PE24 (см. OASIS PE:2006) предлагает заменить вышеприведенный параграф следующим текстом:

Подобно SOAP, URI разрешение может выполняться для многих нижележащих средств транспорта. Эта связь имеет аспекты, не зависящие от протокола, но также предусматривает обязательную реализацию HTTP идентификаторов URI.

#### 10.2.7.1 Требуемая информация

**Идентификация**: `urn:oasis:names:tc:SAML:2.0:bindings:URI`

**Контактная информация**: security-services-comment@lists.oasis-open.org

**Описание**: Приводится ниже.

**Обновления**: Нет.

#### 10.2.7.2 Аспекты связи SAML URI, не зависящие от протокола

В последующих разделах определяются аспекты связи SAML URI, которые не зависят от нижележащего транспортного протокола процесса разрешения URI.

Ссылка SAML URI идентифицирует конкретное подтверждение SAML. Результатом превращения ссылки URI в абсолютную форму URI должно быть сообщение, содержащее подтверждение, или ошибка, зависящая от транспорта. Конкретный формат сообщения зависит от используемого транспортного протокола. Если транспортный протокол позволяет, чтобы возвращенное содержание было бы описано как HTTP 1.1, то подтверждение может быть закодировано в любом разрешенном формате. Если нет, то подтверждение должно быть возвращено в форме, которая может быть однозначно интерпретирована или преобразована в серию подтверждений XML.

Это должно произойти, если такая же ссылка URI будет разрешаться в будущем, тогда возвращается либо то же самое подтверждение SAML, либо ошибка. То есть ссылка может быть постоянной, но должна неизменно ссылаться на одно и то же подтверждение (если таковое есть).

#### 10.2.7.3 Аспекты безопасности

Непрямое использование подтверждения SAML представляет собой опасность, если небезопасна связь ссылки с результатом. Конкретные угрозы и их серьезность зависят от варианта использования подтверждения. Как правило, результату преобразования ссылки на URI в подтверждение SAML можно доверять только, если запрашивающая сторона может быть уверена в идентификации отвечающей стороны и в том, что содержание не было изменено в процессе передачи.

Зачастую недостаточно, чтобы было подписано само подтверждение, поскольку ссылки URI по своей природе несколько скрыты от запрашивающей стороны. Запрашивающая сторона должна обладать независимыми средствами для гарантии того, что возвращенное подтверждение, в действительности, является тем, которое представлено при помощи URI; это выполняется на основании, как аутентификации отвечающей стороны, так и целостности ответа.

#### 10.2.7.4 Инкапсуляция MIME

Для протоколов разрешения, поддерживающих MIME как описание содержания и механизм упаковки, результирующее подтверждение должно быть возвращено как блок MIME типа `application/amlassertion+xml`, как определено в Дополнение II.

#### 10.2.7.5 Использование URI протокола HTTP

Ответственный орган SAML, который требует соответствия связи URI языку SAML, должен обеспечить поддержку HTTP. В настоящем разделе описываются некоторые аспекты использования идентификаторов URI протокола передачи гипертекста (HTTP), включая синтаксис URI, заголовки HTTP и сообщение об ошибках.

##### 10.2.7.5.1 Синтаксис URI

Как правило, не существует ограничений допустимого синтаксиса SAML для ссылок URI до тех пор, пока орган SAML, ответственный за ссылку, создает сообщение, содержащее ее. Однако ответственные органы должны поддерживать оконечную точку URL, на которую запрос HTTP может быть передан с одним-единственным параметром строки запроса, имеющим название `ID`. В оконечной точке URL не должно быть строки запроса, независимой от этого параметра.

Например, если задокументированная оконечная точка у ответственного органа имеет вид "https://saml.пример.edu/подтверждения", то запрос подтверждения с `ID = abcde` может быть передан на:

```
https://saml.example.edu/assertins?ID=abcde
```

Для таких запросов ID не разрешено использовать пробелы.

ПРИМЕЧАНИЕ (информативное). – PE31 (см. OASIS PE:2006) предлагает заменить вышеприведенный текст следующим:

Отметим, что синтаксис URI не поддерживает использование пробелов в таких запросах.

#### 10.2.7.5.2 HTTP и аспекты кэширования

Прокси–элементы HTTP не должны кэшировать подтверждения SAML. Для обеспечения этого должны выполняться следующие правила.

Возвращая подтверждения SAML с использованием HTTP 1.1, отвечающие стороны HTTP должны:

- включить поле заголовка `Cache-Control`, установленное в значение "`no-cache, no-store`";
- включить поле заголовка `Pragma`, установленное в значение "`no-cache`".

#### 10.2.7.5.3 Аспекты безопасности

IETF RFC 2617 описывает возможные атаки в среде HTTP, где используются базовые схемы аутентификации или схемы аутентификации на основании сжатого сообщения.

Настоятельно рекомендуется использовать TLS 1.0 в качестве средства аутентификации, защиты целостности и обеспечения конфиденциальности.

#### 10.2.7.5.4 Сообщение об ошибках

Как и в ходе протокольного обмена HTTP должен использоваться соответствующий код статуса HTTP для указания результата. Например, отвечающая сторона SAML, которая отказывается выполнять обмен сообщениями с запрашивающей стороной SAML, должна вернуть ответ "`403 Forbidden`". Если подтверждение определено как неизвестное для отвечающей стороны, то должен быть возвращен ответ "`404 Not Found`". В таких случаях содержание тела сообщения HTTP не имеет значения.

#### 10.2.7.5.5 Аспекты метаданных

Поддержку связи URI поверх HTTP следует обозначить, указав оконечные точки URL в которые должны быть переданы запросы и ответы.

#### 10.2.7.5.6 Пример обмена сообщениями SAML с использованием URI протокола HTTP

Далее показан пример запроса подтверждения.

```
GET /SamlService?ID=abcde HTTP/1.1
Host: www.example.com
```

Далее показан пример соответствующего ответа, который представляет запрошенное подтверждение.

```
HTTP/1.1 200 OK
Content-Type: application/samlassertion+xml
Cache-Control: no-cache, no-store
Pragma: no-cache
Content-Length: nnnn

<saml:Assertion ID="abcde" ...>
...
</saml:Assertion>
```

## 11 Профили для SAML

В настоящем разделе определяются профили, которые определяют использование подтверждения SAML и сообщения запрос-ответ в протоколах и системах связи, а также профили, которые определяют синтаксис значения атрибута SAML и названия условных обозначений.

### 11.1 Концепции профиля

Один из типов профилей SAML объединяет множество правил, описывающих, как вводить в рамки протокола подтверждения SAML и извлекать их оттуда. Такой профиль описывает, как сторона, создающая подтверждения SAML, может вводить их в другие объекты (например, файлы различных типов или протокольные блоки данных протоколов связи) или объединять с ними, как эти подтверждения могут передаваться от создающей стороны принимающей стороне, и затем обрабатываться в точке назначения. Конкретный набор правил по введению подтверждений SAML в определенный класс объектов`<FOO>` и извлечению их оттуда называется *<FOO> профилем языка SAML.*

Например, профиль SOAP языка SAML описывает, как подтверждения SAML могут быть добавлены в SOAP сообщения, как изменяются заголовки SOAP под влиянием подтверждений SAML, и как в сообщениях SOAP должны обозначаться ошибки, связанные с SAML.

Другой тип профиля SAML определяет набор ограничений, накладываемых на использование общего протокола SAML или возможности создания подтверждений для конкретных условий или используемого контекста. Профили такого типа могут ограничивать функциональные возможности, требовать использования определенных функциональных возможностей SAML (например, атрибутов, условий или связей) и в других аспектах определять правила обработки, которые должны программными средствами данного профиля.

Конкретными примерами последнего случая являются примеры, касающиеся атрибутов SAML. Элемент SAML `<Attribute>` обеспечивает большую гибкость в присваивании имен атрибутам, синтаксисе величин, и введении метаданных за счет использования атрибутов XML. Возможности взаимодействия достигаются за счет ограничения этой гибкости, когда это оправдано, путем четкого следования профилям, которые определяют, как использовать эти элементы при более серьезных ограничениях, чем те, что наложены общими правилами, определенными в разделе 8.

Профили атрибута содержат определения, необходимые для ограничения выражения атрибутов SAML при работе с определенными типами информации атрибутов или при взаимодействии с внешними системами или открытыми стандартами, требующими большей точности.

Предназначение настоящей Рекомендации заключается в том, чтобы определить конкретный набор профилей различного типа настолько подробно, чтобы обеспечить возможность взаимодействия независимо реализованных продуктов.

## 11.2 Спецификация дополнительных профилей

Настоящая Рекомендация определяет конкретный набор профилей, но в будущем, возможно, будут разработаны новые профили. Приведенные далее подклассы являются руководством для определения профилей.

### 11.2.1 Руководство для определения профилей

В настоящем разделе приведен контрольный список вопросов, которые должны быть рассмотрены для каждого профиля.

1) Определить URI, который уникальным образом идентифицирует профиль, почтовые и электронные контактные данные его автора, и дает ссылки на ранее определенные профили, которые новый профиль обновляет или заменяет.
2) Описать набор операций взаимодействия между сторонами, вовлеченными в данный профиль. Должны быть явно названы любые ограничения на приложения, используемые каждой стороной и протоколы, используемые при каждом взаимодействии.
3) Определить стороны, участвующие в каждом взаимодействии, включая то, сколько сторон участвует, и могут ли привлекаться промежуточные элементы.
4) Определить способ аутентификации сторон, участвующих в каждом взаимодействии, включая то, требуется ли аутентификация, и каковы приемлемые типы аутентификации.
5) Определить уровень поддержки целостности сообщения, включая механизмы, используемые для обеспечения целостности сообщения.
6) Определить уровень поддержки конфиденциальности, включая то, может ли третья сторона видеть содержание сообщений и подтверждений SAML, требуется ли конфиденциальность профиля, и какие механизмы рекомендованы для обеспечения конфиденциальности.
7) Определить состояния ошибки, включая состояния ошибки на стороне каждого участника, особенно тех, кто принимает и обрабатывает подтверждения или сообщения SAML.
8) Определить аспекты безопасности, включая анализ угроз и описание контрмер.
9) Определить идентификаторы SAML метода подтверждения, определенные и/или используемые профилем.
10) Определить соответствующие метаданные SAML, определенные и/или используемые профилем.

### 11.2.2 Руководство для определения профилей атрибутов

В настоящем разделе приведен контрольный список вопросов, которые должны быть, в частности, рассмотрены для профилей атрибутов.

1) Определить URI, который уникальным образом идентифицирует профиль, почтовые и электронные контактные данные его автора, и дает ссылки на ранее определенные профили, которые новый профиль обновляет или заменяет.
2) Синтаксис и ограничения допустимых значений атрибутов `NameFormat` и `Name` элементов SAML `<Attribute>`.
3) Любые дополнительные атрибуты XML, определенные профилем в области имен, которые могут использоваться в элементах SAML `<Attribute>`.
4) Правила для определения равенства элементов SAML `<Attribute>`, определенных в профиле, предназначенные для использования при обработке атрибутов, запросов и т. д.
5) Синтаксис и ограничения допустимых значений элемента SAML `<AttributeValue>`, включая то, может ли или должен ли использоваться атрибут XML `xsi:type`.

### 11.3 Идентификаторы метода подтверждения

В разделе 8 определяется элемент <SubjectConfirmation> как атрибут Method плюс дополнительные данные <SubjectConfirmationData>. Элемент <SubjectConfirmation> должен использоваться доверяющей стороной для подтверждения того, что запрос или сообщение получено от элемента системы, который ассоциирован с объектом подтверждения в рамках определенного профиля.

Атрибут Method указывает конкретный метод, который должна использовать доверяющая сторона для выполнения этого определения. Все это может иметь или не иметь отношения к аутентификации, которая была выполнена ранее. В отличие от аутентификации контекста, объект метода подтверждения часто будет сопровождаться дополнительной информацией, такой как сертификат или ключ в элементе <SubjectConfirmationData>, которая позволит доверяющей стороне выполнить необходимые проверки. Кроме того, определяется общий набор атрибутов, который может использоваться для ограничения условий, при которых может выполняться проверка.

Предполагается, что профили будут определять и использовать несколько различных значений атрибута <ConfirmationMethod>, каждое из которых будет соответствовать различным сценариям использования SAML. Приведенные далее методы определены для использования профилями, описанными в настоящей Рекомендации, и другими профилями, для которых они могу быть полезными.

#### 11.3.1 Владелец ключа

**URI:** urn:oasis:names:tc:SAML:2.0:cm:holder-of-key

В элементе <SubjectConfirmationData> должен быть представлен один или несколько элементов <ds:KeyInfo>. Атрибут xsi:type может быть представлен в элементе <SubjectConfirmationData> и, если представлен, должен быть установлен в значение **saml:KeyInfoConfirmationDataType** (префикс области имен произволен, но должен ссылаться на область имен подтверждения SAML).

Как описано в Правилах подписи W3C, каждый элемент <ds:KeyInfo> содержит ключ или информацию, которая позволяет приложению получить ключ. Владелец определенного ключа считается доверяющей стороной объектом подтверждения.

В соответствии с Правилами подписи W3C каждый элемент <ds:KeyInfo> должен определить один-единственный ключ шифрования. В различных элементах <ds:KeyInfo> могут быть определены различные ключи, так например, когда для различных доверяющих сторон требуются различные ключи подтверждения.

**Пример**: Владелец ключа, имеющий название "By-Tor" или владелец ключа, имеющий название "Snow Dog" могут подтвердить сами себя как предмет подтверждения.

```
<SubjectConfirmation Method="urn:oasis:names:tc:SAML:2.0:cm:holder-of-key">
 <SubjectConfirmationData xsi:type="saml:KeyInfoConfirmationDataType">
    <ds:KeyInfo>
       <ds:KeyName>By-Tor</ds:KeyName>
    </ds:KeyInfo>
    <ds:KeyInfo>
       <ds:KeyName>Snow Dog</ds:KeyName>
    </ds:KeyInfo>
 </SubjectConfirmationData>
</SubjectConfirmation>
```

#### 11.3.2 Отправитель подтверждений

**URI:** urn:oasis:names:tc:SAML:2.0:cm:sender-vouches

Указывает, что нет другой доступной информации относительно контекста использования подтверждения. Доверяющая сторона должна использовать другие средства для определения того, должна ли она далее обрабатывать подтверждение, в зависимости от возможных ограничений на подтверждения, использующие атрибуты, которые могут быть представлены в элементе <SubjectConfirmationData>.

#### 11.3.3 Канал передачи информации

**URI:** urn:oasis:names:tc:SAML:2.0:cm:bearer

Объектом подтверждения является канал передачи информации подтверждения, в зависимости от возможных ограничений на подтверждения, использующие атрибуты, которые могут быть представлены в элементе <SubjectConfirmationData>, как определено в разделе 8.

**Пример**: Канал передачи информации может подтвердить сам себя как предмет подтверждения, при условии, что подтверждение доставляется в сообщении, переданном по адресу "https://www.serviceprovider.com/saml/consumer" до 1:37 PM GMT 19 марта 2004 года, в ответ на запрос имеющий ID "_1234567890".

```
<SubjectConfirmation Method="urn:oasis:names:tc:SAML:2.0:cm:bearer">
 <SubjectConfirmationData InResponseTo="_1234567890"
    Recipient="https://www.serviceprovider.com/saml/consumer"
    NotOnOrAfter="2004-03-19T13:27:00Z"
 </SubjectConfirmationData>
</SubjectConfirmation>
```

### 11.4 Профили SSO языка SAML

Для поддержания возможности Единой регистрации с однократным вводом пароля (SSO) для браузеров и других клиентских устройств определен набор профилей.

- Для поддержания возможности Единой регистрации в веб с однократным вводом пароля в разделе 8 определен Профиль протокола запроса аутентификации на базе веб-браузера.
- Для поддержания дополнительных клиентов определен дополнительный профиль SSO на базе веб-браузера.
- Для связей в прямом (браузеры) и обратном каналах в разделе 8 определен Профиль протоколов Единого выхода из сети и управления идентификатором имени.
- Для определения идентификации провайдера определен дополнительный профиль, использующий идентификационные маркеры HTTP, сохраняемые в клиентской системе (cookies).

#### 11.4.1 Профиль SSO для веб-браузера

В сценарии, поддерживаемом профилем SSO для веб-браузера, пользователь веб либо получает доступ к ресурсам на базе провайдера услуг, либо доступ к средствам идентификации провайдера, так, чтобы и провайдер услуг, и желаемый ресурс были бы понятны или определены. Пользователь веб аутентифицирует себя (или уже был аутентифицирован ранее) для провайдера идентификации, который затем создает подтверждение аутентификации (возможно, используя данные, полученные от провайдера услуг), а провайдер услуг использует это подтверждение для того, чтобы установить безопасный обмен контекстом с пользователем веб. Во время этого процесса между провайдерами может быть сформирован для данного клиента идентификатор имени, это зависит от параметров взаимодействия и согласия сторон.

Для реализации этого сценария используется профили протокола запроса аутентификации языка SAML совместно со связями перенаправления HTTP, HTTP POST и артефакта HTTP.

Предполагается, что пользователь использует стандартный коммерчески доступный браузер и может аутентифицировать себя для провайдера идентификации при помощи каких-либо средств вне рамок SAML.

##### 11.4.1.1 Требуемая информация

**Идентификация**: `urn:oasis:names:tc:SAML:2.0:profiles:SSO:browser`

**Контактная информация**: security-services-comment@lists.oasis-open.org

**Идентификаторы языка SAML для метода подтверждения**: Этим профилем используется идентификатор языка SAML для метода "канала передачи" "bearer" `urn:oasis:names:tc:SAML:2.0:cm:bearer`.

**Описание**: Приводится ниже.

**Обновления**: Нет.

##### 11.4.1.2 Обзор профиля

На рисунке 11-1 показан базовый шаблон реализации SSO. Профилем описываются следующие этапы. В рамках отдельного этапа может быть организован один или несколько обменов сообщениями в зависимости от связи, используемой для этого этапа и функций, определяемых вариантом реализации.

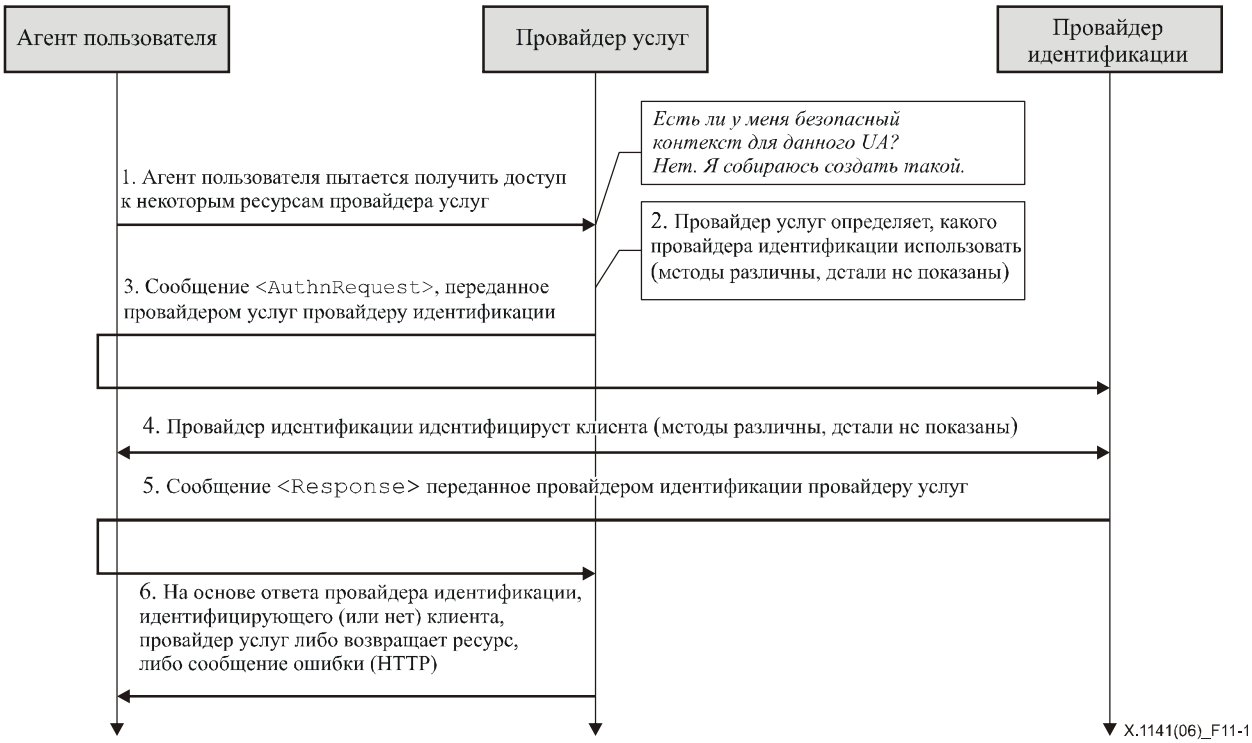


**Рисунок 11-1/X.1141 – Базовый шаблон реализации SSO**

**1) Запрос HTTP провайдеру услуг**

На этапе 1 клиент через агента пользователя HTTP, направляет провайдеру услуг запрос HTTP на безопасные ресурсы, не содержащий контекста безопасности.

**2) Провайдер услуг определяет провайдера идентификации**

На этапе 2 провайдер услуг получает данные о размещении оконечной точки у провайдера идентификации для протокола запроса аутентификации, который поддерживает предпочтительную для него связь. Средства, при помощи которых это выполняется, зависят от варианта реализации. Провайдер услуг может использовать профиль обнаружения провайдера идентификации SAML, описанный в 8.7.4.

**3) Запрос <AuthnRequest>, переданный провайдером услуг провайдеру идентификации**

На этапе 3 провайдер услуг передает сообщение `<AuthnRequest>`, которое должно быть доставлено агентом пользователя провайдеру идентификации. Для передачи этого сообщения провайдеру идентификации через агента пользователя может использоваться одна из связей – HTTP Redirect, HTTP POST или HTTP Artifact.

**4) Провайдер идентификации идентифицирует клиента**

На этапе 4 клиент идентифицируется провайдером идентификации при помощи некоторых средств, которые не входят в описание данного профиля. Для этого могут потребоваться новые действия по аутентификации, или может еще раз использоваться существующий сеанс аутентификации.

**5) Провайдер идентификации передает провайдеру услуг сообщение <Response>**

На этапе 5 провайдер идентификации передает сообщение `<Response>`, которое должно быть доставлено агентом пользователя провайдеру услуг. Для передачи этого сообщения провайдеру услуг через агента пользователя может использоваться одна из связей – HTTP POST или Артефакт HTTP. Сообщение может указывать ошибку, или содержать (как минимум) подтверждение аутентификации. Связь HTTP Redirect не должна использоваться, поскольку этот ответ будет обычно превышать длину URL, допускаемую большинством агентов пользователя.

**6) Провайдер услуг разрешает клиенту доступ или отказывает в доступе**

На этапе 6 получив ответ от провайдера идентификации, провайдер услуг может ответить агенту пользователя клиента – либо сообщить об ошибке, либо установить свой собственный контекст безопасности для клиента и вернуть запрошенный ресурс.

Провайдер идентификации может инициировать выполнение этого профиля на этапе 5 и передать провайдеру услуг сообщение `<Response>` без предыдущих этапов.

#### 11.4.1.3 Описание профиля

Если профиль инициируется провайдером услуг, начинайте с 11.4.1.3.1. Если профиль инициируется провайдером идентификации, начинайте с 11.4.1.3.5. В нижеприведенных описаниях используются следующие обозначения:

**Служба Единой регистрации с однократным вводом пароля**

Это оконечная точка протокола запроса аутентификации на стороне провайдера идентификации, на которую агент пользователя доставляет сообщение `<AuthnRequest>` (или представляющий его артефакт).

**Служба Подтверждения пользователя**

Это оконечная точка протокола запроса аутентификации на стороне провайдера провайдер услуг, на которую агент пользователя доставляет сообщение `<Response>` (или представляющий его артефакт).

##### 11.4.1.3.1 Запрос HTTP провайдеру услуг

Если это первый доступ к провайдеру услуг, то профиль может быть инициирован произвольным запросом ресурсов. Не существует никаких ограничений на форму этого запроса. Провайдер услуг может использовать любые желаемые средства для связывания последовательных взаимодействий с исходным запросом. Каждая связь предоставляет механизм RelayState, который провайдер услуг может использовать для связи действий обмена данными, предусмотренных профилем, с исходным запросом. Провайдер услуг в значении `RelayState` должен минимально возможно раскрывать содержание запроса, если только использования этого профиля не требует применения мер по обеспечению секретности.

##### 11.4.1.3.2 Провайдер услуг определяет провайдера идентификации

Этот этап зависит от варианта реализации. Провайдер услуг может использовать профиль обнаружения провайдера идентификации языка SAML, описанный в 11.4.3. Провайдер услуг может также принять решение о перенаправлении агента пользователя на другую службу, которая имеет возможность определить соответствующего провайдера идентификации. В таком случае провайдер услуг может передать на эту службу запрос `<AuthnRequest>` (как в следующем этапе), который должен быть ретранслирован провайдеру идентификации, или может быть ретранслирован на промежуточную службу для создания от ее имени сообщения `<AuthnRequest>`.

##### 11.4.1.3.3 Сообщение <AuthnRequest>, передаваемое провайдером услуг провайдеру идентификации

После того как провайдер идентификации выбран, определяется местоположение его Службы Единой регистрации с однократным вводом пароля, на основании связи SAML, выбранной провайдером услуг для передачи сообщения `<AuthnRequest>`. Для этой цели могут использоваться метаданные. В ответ на запрос HTTP от агента пользователя, возвращается ответ HTTP содержащий сообщение `<AuthnRequest>` или артефакт, в зависимости от используемой связи SAML, которое должно быть доставлено на службу Единой регистрации провайдера идентификации.

Точный формат этого ответа HTTP и последующего запроса HTTP в службу Единой регистрации определяется используемой связью SAML. Правила, определяемые профилем для содержания сообщения `<AuthnRequest>`, перечислены в 11.4.1.4.1. Если используется связь HTTP Redirect или HTTP POST, то сообщение `<AuthnRequest>` на этом этапе доставляется непосредственно провайдеру идентификации. Если используется связь HTTP Artifact, то профиль разрешения артефакта, определенный в 11.4.6 используется провайдером идентификации, который выполняет обратный вызов в отношении провайдера услуг для получения сообщения `<AuthnRequest>`, используя, например, связь SOAP.

Рекомендуется, чтобы все обмены данными HTTP на этом этапе выполнялись по TLS 1.0 для сохранения конфиденциальности и целостности сообщения. Если требуется аутентификация создателя запроса, сообщение `<AuthnRequest>` может быть подписано. Связь HTTP Artifact, если она используется, также предоставляет дополнительные средства аутентификации создателя запроса, когда обозначен артефакт.

Провайдер идентификации должен обработать сообщение `<AuthnRequest>`, как описано в настоящей Рекомендации. Это может ограничить последующее взаимодействия с агентом пользователя, например, если включен атрибут `IsPassive`.

##### 11.4.1.3.4 Провайдер идентификации идентифицирует клиента

В любой момент в ходе предыдущего или последующего этапа, провайдер идентификации должен установить идентификацию клиента (если только он не возвращает провайдеру услуг сообщение об ошибке). Атрибут `ForceAuthn` `<AuthnRequest>`, если он представлен со значением "true", обязывает провайдера идентификации заново установить идентификацию, а не полагаться на тот сеанс связи, который у него может существовать с клиентом. В противном случае во всех других аспектах провайдер идентификации может использовать любые средства для того, чтобы аутентифицировать агента пользователя, в зависимости от требований, содержащихся в запросе `<AuthnRequest>` в виде элемента `<RequestedAuthnContext>`.

##### 11.4.1.3.5 Провайдер идентификации передает <Response> провайдеру услуг

Вне зависимости от успешного или безуспешного результата обработки `<AuthnRequest>`, провайдер идентификации должен передать агенту пользователя ответ HTTP, содержащий сообщение `<Response>` или артефакт, в зависимости от используемой связи SAML, которые должны быть доставлены службе подтверждения пользователей провайдера услуг.

Точный формат этого ответа HTTP и последующего запроса HTTP в службу подтверждения пользователей определяется используемой связью SAML. Определяемые профилем правила для содержания сообщения <Response>, перечислены в 11.4.1.4.2. Если используется связь HTTP POST, то сообщение <Response> на этом этапе доставляется непосредственно провайдеру услуг. Если используется связь HTTP Artifact, то профиль разрешения артефакта, определенный в 11.4.6 используется провайдером услуг, который выполняет обратный вызов в отношении провайдера идентификации для получения сообщения <Response>, используя, например, связь SOAP.

Местоположение службы подтверждения пользователей может быть определено с использованием метаданных. Провайдер идентификации должен иметь средства для определения, что это местоположение в действительность, подконтрольно провайдеру услуг. Провайдер услуг в своем запросе<AuthnRequest> может указать, какие связь SAML и конкретную службу подтверждения пользователей следует использовать, и провайдер идентификации должен, по возможности, учесть эти требования.

Рекомендуется, чтобы запросы HTTP на этом этапе выполнялись по TLS 1.0 для сохранения конфиденциальности и целостности сообщения. Элемент(ы) <Assertion> в ответе <Response> должны быть подписаны, если используется связь HTTP POST, и могут быть подписаны, если используется связь HTTP Artifact.

Провайдер услуг должен обработать сообщение <Response> и все вложенные элементы <Assertion>, как описано в настоящей Рекомендации.

#### 11.4.1.3.6 Провайдер услуг разрешает агенту пользователя доступ или отказывает в доступе

Для завершения профиля, провайдер услуг обрабатывает сообщения <Response> и <Assertion> и разрешает доступ к ресурсу или отказывает в таком доступе. Провайдер услуг может установить с агентом пользователя контекст безопасности, используя любой механизм сеанса связи по своему выбору. Все последующие случаи использования представленного(ых) сообщения(й) <Assertion> остаются на усмотрение провайдера услуг и других доверяющих сторон, с учетом содержащихся в них ограничений на использования.

### 11.4.1.4 Использование протокола запроса аутентификации

Этот профиль основан на протоколе запроса аутентификации, определенном в настоящей Рекомендации. Здесь, провайдер услуг является создателем запроса и доверяющей стороной, а клиент представляет собой объект запроса и подтверждающую сторону. По усмотрению провайдера идентификации могут существовать дополнительные доверяющие или подтверждающие стороны.

#### 11.4.1.4.1 Использование запроса <AuthnRequest>

Провайдер услуг может включить в запрос любой содержание сообщения, описанное в настоящей Рекомендации. Все используемые правила обработки определены в настоящей Рекомендации. В запросе должен быть представлен элемент <Issuer>, и он должен содержать уникальный идентификатор запрашивающего провайдера услуг; атрибут Format должен быть пропущен или должен иметь значение urn:oasis:names:tc:SAML:2.0:nameid-format:entity.

Если провайдер идентификации не может или не будет выполнять запрос, он должен ответить, передавая сообщение <Response>, содержащее соответствующий(е) код(ы) статуса ошибки.

Если провайдер услуг желает разрешить провайдеру идентификации создать новый идентификатор для клиента, если такового пока не существовало, он должен содержать элемент <NameIDPolicy> с атрибутом AllowCreate, имеющим значение "true". В противном случае успешно может быть аутентифицирован только тот клиент, для которого провайдер идентификации ранее создал идентификатор, который может быть использован данным провайдером услуг.

Провайдер услуг может включить в запрос элемент <Subject>, который обозначает действительный идентификатор, для которого он желает пучить подтверждение. Этот элемент не должен содержать элементов <SubjectConfirmation>. Если провайдер идентификации не распознает клиента, как имеющего этот идентификатор, то он должен ответить, передавая сообщение <Response>, содержащее статус ошибки, а не подтверждение.

Сообщение <AuthnRequest> может быть подписано (как указывается используемой связью SAML). Если используется связь HTTP Artifact, аутентификация сторон является дополнительной и может использоваться любой механизм, разрешенный данной связью.

Если запрос <AuthnRequest> не аутентифицируется и его целостность не защищается, то информация в нем не должна считаться достоверной, а только рекомендательной. Вне зависимости от того, подписан или нет, этот запрос, провайдер идентификации должен гарантировать, что для любых элементов <AssertionConsumerServiceURL> или <AssertionConsumerServiceIndex> в запросе подтверждена их принадлежность провайдеру услуг, на запрос которого будет направлен этот ответ. Невозможность сделать это может привести к атаке с третьей стороны.

#### 11.4.1.4.2 Использование ответа <Response>

ПРИМЕЧАНИЕ 1 (информативное). – PE26 (см. OASIS PE:2006) предлагает разъяснить предназначение данного подраздела, подробности приведены в Дополнении VIII.

Если провайдер идентификации желает вернуть ошибку, он не должен включать в сообщение <Response> каких-либо подтверждений. В противном случае если запрос выполнен успешно (или если ответ никак не связан с запросом), элемент <Response> должен соответствовать нижеследующему описанию.

- Элемент <Issuer> может быть пропущен, но, если он представлен, он должен содержать уникальный идентификатор создавшего его провайдера идентификации; атрибут Format должен быть пропущен или должен иметь значение urn:oasis:names:tc:SAML:2.0:nameid-format:entity.

  ПРИМЕЧАНИЕ 2 (информативное). – PE17 (см. OASIS PE:2006) предлагает заменить вышеприведенный параграф следующим:

  Если сообщение <Response> подписано, или если вложенное подтверждение зашифровано, то элемент <Issuer> должен быть представлен. В противном случае он может быть пропущен. Если он представлен, он должен содержать уникальный идентификатор создавшего его провайдера идентификации; атрибут Format должен быть пропущен или должен иметь значение urn:oasis:names:tc:SAML:2.0:nameid-format:entity.
- Он должен содержать как минимум одно подтверждение <Assertion>. Элемент <Issuer> каждого подтверждения должен содержать уникальный идентификатор создавшего его провайдера идентификации; атрибут Format должен быть пропущен или должен иметь значение urn:oasis:names:tc:SAML:2.0:nameid-format:entity.
- Множество из одного или нескольких подтверждений должно содержать как минимум один элемент <AuthnStatement>, который отражает аутентификацию клиента для провайдер идентификации.
- Элемент <Subject> должен содержать как минимум одно утверждение, содержащее элемент <AuthnStatement>, в котором имеется как минимум один элемент <SubjectConfirmation>, содержащий Method = urn:oasis:names:tc:SAML:2.0:cm:bearer. Если провайдер идентификации поддерживает профиль единого выхода из системы (Single Logout profile), определенный в 11.4.4, то любое такое утверждение аутентификации должно содержать атрибут Index сеанса связи, который позволяет провайдеру услуг создавать запрос выхода из системы отдельно для каждого сеанса связи.
- Вышеописанный элемент канала передачи <SubjectConfirmation> должен содержать элемент <SubjectConfirmationData>, который содержит атрибут Recipient, содержащий URL службы подтверждения пользователя провайдера услуг, и атрибут NotOnOrAfter, который ограничивает временной интервал, в течение которого может быть доставлено подтверждение. Он может содержать атрибут Address, ограничивающие клиентские адреса, с которых быть доставлено подтверждение. Он не должен содержать атрибута NotBefore. Если сообщение, содержащее этот элемент, представляет собой ответ на запрос <AuthnRequest>, то атрибут InResponseTo должен соответствовать ID запроса.
- Другие утверждения и методы подтверждения могут включаться в подтверждение(я) по усмотрению провайдера идентификации. В частности, могут включаться элементы <AttributeStatement>. Запрос <AuthnRequest> может содержать XML атрибут ConsumingServiceIndex, обозначая информацию о желаемых или требуемых атрибутах, описанных в разделе 9. Провайдер идентификации может игнорировать эти данные, или передавать другие атрибуты по своему усмотрению.
- Подтверждение(я), содержащее(ие) подтверждение объекта канала передачи, должно(ы) содержать элемент <AudienceRestriction>, включая уникальный идентификатор провайдера услуг в виде элемента <Audience>.
- Другие условия (и другие элементы <Audience>) могут быть включены по запросу провайдера услуг или по усмотрению провайдера идентификации. (Несомненно, все такие условия должны быть понятны и приемлемы для провайдера услуг для того, чтобы подтверждение могло считаться достоверным.) Провайдер идентификации не обязан учитывать запрошенный набор условий <Conditions> в запросе <AuthnRequest>, если таковые указаны.

#### 11.4.1.4.3 Правила обработки сообщения <Response>

ПРИМЕЧАНИЕ (информативное). – PE26 (см. OASIS PE:2006) предлагает разъяснить предназначение данного подраздела, подробности приведены в Дополнении VIII.

Вне зависимости от используемой связи SAML, провайдер услуг должен выполнить следующее:

- проверить все подписи, представленные в подтверждении(ях) или ответе;
- проверить, что атрибут получателя в любом <SubjectConfirmationData> канала передачи соответствует URL службы подтверждения пользователя, которой был доставлен <Response> или артефакт;
- проверить, что атрибут NotOnOrAfter не передается ни в одном <SubjectConfirmationData> канала передачи, учитывая допустимый сдвиг синхронизации между провайдерами;
- проверить, что атрибут InResponseTo в <SubjectConfirmationData> канала равен ID исходного сообщения <AuthnRequest>, если только ответ не является сообщением без запроса, в котором этот атрибут не должен быть представлен;
- проверить, что все используемые подтверждения достоверны в других отношениях;
- если какой-либо элемент <SubjectConfirmationData> канала передачи содержит атрибут Address, провайдер услуг может сверить с ним клиентский адрес агента пользователя;

- любое подтверждение, которое не является достоверным, или для субъекта которого не могут быть выполнены требования по подтверждению, должно быть отброшено и не должно использоваться для установления контекста безопасности для клиента;
- если утверждение <AuthnStatement>, используемое для установления контекста безопасности для клиента, содержит атрибут сеанса связи NotOnOrAfter, то контекст безопасности должен быть отброшен сразу по достижении этого времени, если только провайдер услуг не установит заново идентификацию клиента, повторив использование этого профиля.

#### 11.4.1.4.4 Правила обработки сообщения <Response>, свойственных связи артефакта

Если для доставки сообщения <Response> используется связь HTTP Artifact, то ссылка на артефакт с применением профиля разрешения артефакта должна быть взаимно аутентифицирована, конфиденциальна и ее целостность должна быть защищена.

Провайдер идентификации должен гарантировать, что только провайдеру услуг, для которого было передано сообщение <Response>, получит данное сообщение, как результат запроса <ArtifactResolve>.

Для аутентификации сторон и защиты сообщений может использоваться либо связь SAML, применяемая для ссылки на артефакт, либо подписи сообщения.

#### 11.4.1.4.5 Правила обработки сообщений, свойственных связи POST

ПРИМЕЧАНИЕ (информативное). – PE26 (см. OASIS PE:2006) предлагает разъяснить предназначение данного подраздела, подробности приведены в Дополнении VIII.

Если связь HTTP POST используется для доставки сообщения <Response>, вложенное(ые) внего подтверждение(я) должно(ы) быть подписано(ы).

Провайдер услуг должен гарантировать, что канал передачи подтверждений не передает их повторно, сохраняя множество использованных значений ID на протяжении всего времени, пока подтверждение может считаться достоверным на основе данных атрибута NotOnOrAfter в элементе <SubjectConfirmationData>.

### 11.4.1.5 Незапрашиваемые ответы

Провайдер идентификации может инициировать этот профиль, доставив провайдеру услуг незапрошенное сообщение <Response>.

Незапрашиваемый ответ <Response> не должен содержать атрибута InResponseTo, и такого атрибута не должен содержать ни один элемент канала передачи <SubjectConfirmationData>. Если используются метаданные, то сообщение <Response> или артефакт должен быть доставлен на оконечную точку провайдера услуг <md:AssertionConsumerService>, которая обозначена как точка "по умолчанию".

Отдельно отмечается, что провайдер идентификации может включить свойственный данному типу связи параметр "RelayState", который основан на взаимном соглашении с провайдером услуг о том, как выполнить последующие взаимодействия с агентом пользователя. Это может быть URL ресурса на стороне провайдера услуг. Провайдер услуг должен быть готов обработать незапрошенные ответы, обозначив точку "по умолчанию", куда должен быть направлен агент пользователя для успешной обработки ответа.

### 11.4.1.6 Использование метаданных

В 11.4.2.5 определяется элемент оконечной точки <md:SingleSignOnService>, описывающий поддерживаемые связи и точку(и), в которые провайдер услуг может направлять запросы провайдеру идентификации, используя этот профиль.

Атрибут WantAuthnRequestsSigned элемента <md:IDPSSODescriptor> может использоваться провайдером идентификации для регистрации того, что запросы были подписаны. Атрибут AuthnRequestsSigned элемента <md:SPSSODescriptor> может использоваться провайдером услуг для указания намерения подписать все свои запросы.

Провайдеры могут записывать ключ(и), используемы(е) для подписания запросов, ответов и подтверждений, применяя элементы <md:KeyDescriptor> с атрибутом use = sign. При кодировании элементов SAML элементы <md:KeyDescriptor> с атрибутом use = encrypt могут использоваться для записи поддерживаемых алгоритмов и установок шифрования, а также открытых ключей, используемых для приема ключей шифрования.

Индексированный элемент оконечной точки <md:AssertionConsumerService> используется для описания поддерживаемой(ых) связи(ей) и точки(ек), на которые провайдер идентификации может передавать ответы провайдеру услуг, используя этот профиль. Атрибут index используется для различения возможных оконечных точек, которые могут быть указаны ссылкой в сообщении <AuthnRequest>. Атрибут isDefault используется для определения оконечной точки, которая будет использована, если в запросе не указана.

Атрибут WantAssertionsSigned элемента <md:SPSSODescriptor> может использоваться провайдером услуг для регистрации требования, чтобы подтверждения, доставленные с этим профилем, были подписаны. Это требование является дополнением любым другим требованиям по подписанию, определенным в результате применения конкретной связи. Это не налагает никаких обязательств на провайдера идентификации, но он информирован о вероятности того, что неподписанного подтверждения будет недостаточно.

Если запрос или ответ доставляется с использованием связи HTTP Artifacta, то создатель артефакта должен сообщить как минимум один элемент оконечной точки <md:ArtifactResolutionService> в своих метаданных.

Дескриптор <md:IDPSSODescriptor> может содержать элементы <md:NameIDFormat>, <md:AttributeProfile> и <saml:Attribute> для указания общей способности поддерживать конкретные форматы идентификатора имени, профили атрибутов или конкретные атрибуты и их значения. Способность поддерживать любую такую возможность во время данного обмена сообщениями аутентификации зависит от установленных правил и остается на усмотрение провайдера идентификации.

Элемент <md:SPSSODescriptor> может также использоваться для регистрации потребности или желания провайдера иметь атрибуты SAML, которые должны быть доставлены вместе с информацией аутентификации. Реальное включение атрибутов всегда остается на усмотрение провайдера идентификации. Один или несколько элементов <md:AttributeConsumingService> может быть включено в его метаданные, каждый из которых имеет индекс атрибута, позволяющий различать различные услуги, которые могут быть указаны посредством ссылки в сообщении <AuthnRequest>. Атрибут isDefault используется для определения набора требований к атрибутам, предъявляемых "по умолчанию".

### 11.4.2 Профиль расширенной архитектуры "клиент/прокси-сервер" (ECP)

*Расширенная архитектура "клиент/прокси-сервер"* (ECP) – это элемент системы, который знает, как связаться с соответствующим провайдером идентификации, возможно контекстно-зависимым способом, и также поддерживает обратную связь SOAP (PAOS) (см. раздел 10).

Примерный сценарий, который реализуется этим профилем, имеет следующий вид: Клиент, имеющий в своем распоряжении ECP, использует его либо для получения доступа к ресурсу на стороне провайдера услуг, или для доступа к провайдеру идентификации, таким образом, чтобы этот провайдер услуг и желаемый ресурс были бы понятны или определены. Клиент аутентифицирует себя (или уже был аутентифицирован ранее) для провайдера идентификации, который затем создает подтверждение аутентификации (возможно, используя данные, полученные от провайдера услуг). Провайдер услуг затем услуг использует это подтверждение и последовательно устанавливает безопасный обмен контекстом для клиента. Во время этого процесса между провайдерами может быть сформирован для данного клиента идентификатор имени, это зависит от параметров взаимодействия и согласия клиента.

Этот профиль основан на протоколе запроса аутентификации SAML вместе со связью PAOS.

ПРИМЕЧАНИЕ. – Средства, при помощи которых клиент аутентифицирует себя для провайдера идентификации, выходят за рамки SAML.

#### 11.4.2.1 Требуемая информация

**Идентификация**: urn:oasis:names:tc:SAML:2.0:profiles:SSO:ecp (это также является целевой областью имен, назначенной соответствующей схеме профиля ECP в Приложении A).

**Контактная информация**: security-services-comment@lists.oasis-open.org

**SAML идентификаторы метода подтверждения**: Этим профилем используется идентификатор языка SAML для метода "канала передачи" ("bearer") urn:oasis:names:tc:SAML:2.0:cm:bearer.

**Описание**: Приводится ниже.

**Обновления**: Нет.

#### 11.4.2.2 Обзор профиля

Вышеописанный профиль ECP определяет взаимодействие между расширенными клиентами или прокси-серверами и провайдерами услуг и провайдерами идентификации. Это – специфическое применение профиля SSO, описанного в 11.4.1. Если этим профилем не определено иного, если рассматриваемый случай не связан с использованием связей на основе браузеров, то должны выполняться правила, определенные в 11.4.1.

ECP – это клиент или прокси-сервер, который удовлетворяет следующим двум условиям:

- он обладает, или знает, как получить информацию об идентификации провайдера, которого клиент ассоциирует с ECP, который он желает использовать, в рамках взаимодействия с провайдером услуг.

Это позволяет провайдеру услуг направлять запрос аутентификации на ECP без необходимости знать, или узнавать идентификацию соответствующего провайдера (т. е. пропуская этап 2 профиля SSO из 11.4.1).

- он может использовать обратную связь SOAP (PAOS), определенную здесь для запросов и ответов аутентификации.

Это позволяет провайдеру услуг получить подтверждение аутентификации через ECP, к которому в противном случае (т. е. вне рамок контекста немедленного взаимодействия) не обязательно допускает непосредственному обращение и не является постоянно доступным. Это также усиливает преимущества SOAP при применении определенной модели обмена и профиля для реализации взаимодействия. ECP можно рассматривать как промежуточный элемент SOAP между провайдером услуг и провайдером идентификации.

*Расширенным клиентом* может быть браузер или какой-либо другой агент пользователя, который поддерживает функции, описанные в этом профиле. *Расширенным прокси-сервером* может быть промежуточный элемент HTTP, который эмулирует расширенного клиента. Если не объявлено иного, все утверждения, относящиеся к расширенным клиентам, следует считать утверждениями, как о расширенных клиентах, так и прокси-серверах расширенных клиентов.

Поскольку расширенный клиент передает и принимает сообщения в теле запросов и ответов HTTP, у него нет ограничений на размеры протокольных сообщений.

Этот профиль усиливает действие обратной SOAP (PAOS) связи (см. раздел 10). Те, кто реализует этот профиль, должны следовать правилам по указаниям HTTP, определяющим поддержку PAOS, указанную в этой связи, в дополнение к тем требованиям, которые определены в этом профиле. Этот профиль использует блок заголовка PAOS SOAP, передаваемый между отвечающей стороной HTTP и ECP, но не определяет саму PAOS. Этот профиль определяет блок заголовка SOAP, который сопровождает запросы и ответы SAML. Эти блоки заголовков могут, при необходимости, быть объединены с другими блоками заголовков, например с блоком заголовка сообщения SOAP о безопасности, для добавления, при необходимости, возможностей обеспечения безопасности, например цифровой подписи на запросе аутентификации.

Используется два множества блоков заголовков запрос/ответ SOAP: блок заголовка PAOS для общей информации PAOS и определяемые профилем блоки заголовка ECP для передачи информации, относящейся к функциям профиля ECP.

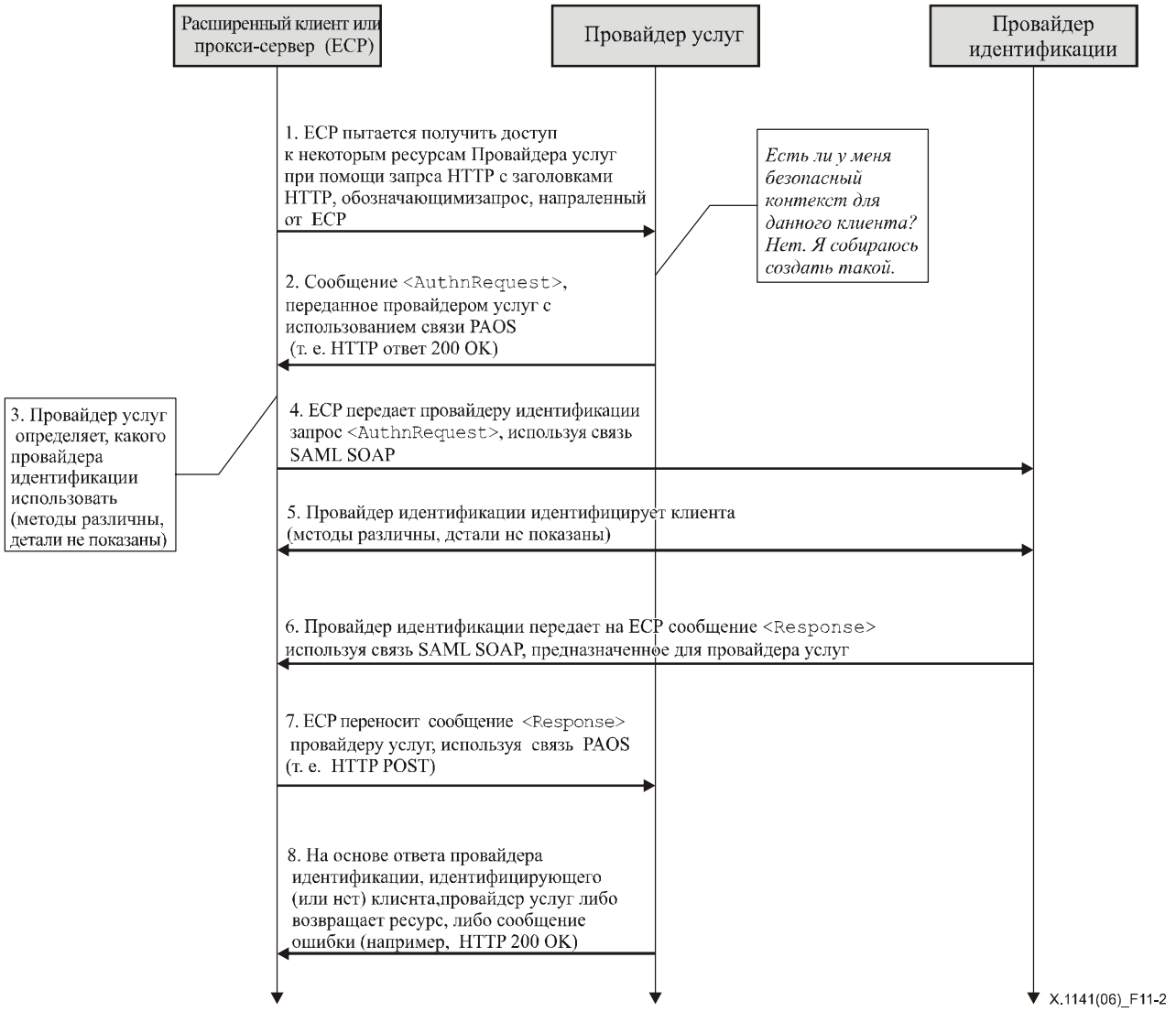


**Рисунок 11-2/X.1141 – Алгоритм обработки в профиле ECP**

На рисунке 11-2 показан базовый шаблон реализации SSO с использованием ECP. Профилем описываются следующие этапы. В рамках отдельного этапа может быть организован один или несколько обменов сообщениями в зависимости от связи, используемой для этого этапа и функций, определяемых вариантом реализации.

**1) ECP передает запрос HTTP провайдеру услуг**

На этапе 1 клиент через ECP, направляет провайдеру услуг запрос HTTP на защищенные ресурсы, для которых провайдер услуг не имеет установленного контекста безопасности для ECP и клиента.

**2) Провайдер услуг передает на ECP сообщение `<AuthnRequest>`**

На этапе 2 провайдер услуг передает на ECP сообщение `<AuthnRequest>`, которое должно быть доставлено при помощи ECP соответствующему провайдеру идентификации. Здесь используется обратная связь SOAP (PAOS) (см. раздел 10).

**3) ECP определяет провайдера идентификации**

На этапе 3 ECP получает данные о размещении оконечной точки у провайдера идентификации для протокола запроса аутентификации, который поддерживает предпочтительную для него связь и получает данные о размещении оконечной точки у провайдера идентификации для протокола запроса. Средства, при помощи которых это выполняется, зависят от варианта реализации. ECP может использовать профиль обнаружения провайдера идентификации SAML, описанный в 11.4.3.

ПРИМЕЧАНИЕ (информативное). – PE18 (см. OASIS PE:2006) предлагает удалить из вышеприведенного параграфа последнюю строчку.

**4) ECP передает запрос `<AuthnRequest>` провайдеру идентификации**

На этапе 4 ECP передает сообщение `<AuthnRequest>` провайдеру идентификации, определенному на этапе 3 с использованием модифицированной формы связи SAML SOAP (см. раздел 10), которое дополнительно разрешает провайдеру идентификации обмениваться произвольными сообщениями HTTP с ECP до ответа на запрос SAML.

**5) Провайдер идентификации идентифицирует клиента**

На этапе 5 клиент идентифицируется провайдером идентификации при помощи некоторых средств, которые не входят в описание данного профиля. Для этого могут потребоваться новые действия по аутентификации, или может еще раз использоваться существующий сеанс аутентификации.

**6) Провайдер идентификации передает на ECP сообщение `<Response>`, предназначенное для провайдера услуг**

На этапе 6 провайдер идентификации с использованием связи SAML SOAP передает сообщение `<Response>`, которое должно быть доставлено провайдеру услуг при помощи ECP. Сообщение может указывать ошибку, или содержать (как минимум) подтверждение аутентификации.

**7) ECP доставляет сообщение `<Response>` провайдеру услуг**

На этапе 7 ECP доставляет провайдеру услуг сообщение `<Response>`, используя PAOS.

**8) Провайдер услуг разрешает клиенту доступ или отказывает в доступе**

На этапе 8 получив сообщение `<Response>` от провайдера идентификации, провайдер услуг либо устанавливает свой собственный контекст безопасности для клиента и возвращает запрошенный ресурс или отвечает клиенту ECP уведомлением об ошибке.

#### 11.4.2.3 Описание профиля

В последующих разделах приводятся подробные определения отдельных этапов.

##### 11.4.2.3.1 ECP передает провайдеру услуг запрос HTTP

ECP передает провайдеру услуг запрос HTTP, определяя некоторые ресурсы. Этот запрос HTTP должен соответствовать связи PAOS, т. е. он должен содержать следующие поля заголовка HTTP:

1) поле заголовка HTTP `Accept`, указывающее способность принять типа MIME "`application/vnd.paos+xml`";
2) поле заголовка HTTP `PAOS`, определяющее версию `PAOS` с как минимум `urn:liberty:paos:2003-08`;
3) кроме того, поддержка этого профиля должна быть указана в поле заголовка HTTP `PAOS` как служебное значение `urn:oasis:names:tc:SAML:2.0:profiles:SSO:ecp`. Это значение должно соответствовать служебному атрибуту в блоке заголовка PAOS запроса SOAP.

Например, агент пользователя может запросить у провайдер услуг страницу следующим образом:

```
GET /index HTTP/1.1
Host: identity-service.example.com
Accept: text/html; application/vnd.paos+xml
PAOS: ver='urn:liberty:paos:2003-08' ;
'urn:oasis:names:tc:SAML:2.0:profiles:SSO:ecp'
```

##### 11.4.2.3.2 Провайдер услуг передает на ECP запрос <AuthnRequest>

Когда провайдер услуг требует наличия контекста безопасности для клиента до того, как разрешает ему доступ к определенным ресурсам, т. е. до предоставления услуг или данных, он может ответить на запрос HTTP, используя связь PAOS с сообщением `<AuthnRequest>` в ответе HTTP. Провайдер услуг передаст на ECP ответ HTTP 200 OK, содержащий одну огибающую SOAP.

Огибающая SOAP должна содержать:

1) элемент <AuthnRequest> в теле SOAP, предназначенный для оконечного получателя SOAP, провайдера идентификации;
2) блок заголовка PAOS SOAP, предназначенный для ECP, использующий значение действия SOAP http://schemas.xmlsoap.org/soap/actor/next. Этот блок заголовка предоставляет управляющую информации, такую как URL, на который будет передан ответ в этом формате обмена запрошенными ответами;
3) блок заголовка запроса SOAP, свойственный профилю ECP, предназначенный для ECP, использующий значение действия SOAP http://schemas.xmlsoap.org/soap/actor/next. Блок заголовка запроса ECP определяет информацию, связанную с запросом аутентификации, которая может потребоваться ECP для его обработки, такую как список провайдеров идентификации, приемлемых для провайдера услуг, вне зависимости от того, может ли ECP взаимодействовать с клиентом, используя понятное человеку имя клиента и провайдера, которое может быть показано клиенту.

Огибающая SOAP может содержать блок SOAP заголовка ECP RelayState, предназначенный для ECP, использующий значение действия SOAP http://schemas.xmlsoap.org/soap/actor/next. Заголовок содержит информацию о состоянии, которая должна быть возвращена при помощи ECP вместе с ответом SAML.

#### 11.4.2.3.3 ECP определяет провайдера идентификации

ECP будет определять, какой провайдер идентификации приемлем и соответствующим образом направлять сообщение SOAP.

#### 11.4.2.3.4 ECP передает запрос <AuthnRequest> провайдеру идентификации

ECP должен удалить блоки заголовка PAOS, ECP RelayState и ECP Request до передачи провайдеру идентификации сообщения <AuthnRequest>, используя модифицированную форму связи SAML SOAP. Запрос SAML передается через SOAP обычным способом, но провайдер идентификации может ответить на запрос HTTP, полученный от ECP, передав ответ HTTP, содержащий, например, форму регистрации HTML или иной ответ, ориентированный на представление. Может иметь место последовательность обменов HTTP, но, в конце концов, провайдер идентификации должен завершить обмен SAML SOAP и вернуть ответ SAML при помощи связи SOAP.

Элемент <AuthnRequest> может быть подписан провайдером услуг. В этом и других смыслах должны выполняться правила для сообщений, определенные в профиле SSO браузера в 11.4.1.4.1.

До или после этого этапа провайдер идентификации должен при помощи некоторых средств установить идентификацию клиента, или он должен возвратить ошибку <Response>, как описано в 11.4.2.3.6.

#### 11.4.2.3.5 Провайдер идентификации идентифицирует клиента

В любой момент в ходе предыдущего или последующего этапа, провайдер идентификации должен установить идентификацию клиента (если только он не возвращает провайдеру услуг сообщение об ошибке). Атрибут ForceAuthn <AuthnRequest>, если он представлен со значением "true", обязывает провайдера идентификации заново установить идентификацию, а не полагаться на тот сеанс связи, который у него может существовать с клиентом. В противном случае во всех других аспектах провайдер идентификации может использовать любые средства для того, чтобы аутентифицировать агента пользователя, в зависимости от требований, содержащихся в запросе <AuthnRequest> в виде элемента <RequestedAuthnContext>.

#### 11.4.2.3.6 Провайдер идентификации передает на ECP <Response>, предназначенный для провайдера услуг

Провайдер идентификации возвращает сообщение SAML <Response> (или ошибка SOAP), когда получает запрос аутентификации, после того как идентификация клиента установлена. Ответ SAML передается с использованием связи SAML SOAP в сообщении SOAP с элементом <Response> в теле SOAP, предназначенном для провайдера услуг, как оконечного получателя SOAP. Правила для этого ответа, определенные в профиле SSO браузера в 11.4.1.4.2, должны выполняться.

Сообщение ответа провайдера идентификации должно содержать блок заголовка ответа SOAP, определяемый профилем ECP, и может содержать блок заголовка ECP RelayState, оба они предназначены для ECP.

#### 11.4.2.3.7 ECP доставляет провайдеру услуг сообщение <Response>

ECP удаляет блок(и) заголовка и может добавить блок заголовка ответа SOAP PAOS и блок заголовка ECP RelayState до перенаправления ответа SOAP провайдеру услуг, используя связь PAOS.

Блок заголовка SOAP <paos:Response> в ответе провайдеру услуг, как правило, используется для корреляции этого ответа с полученным ранее запросом от провайдера услуг. В этом профиле корреляция атрибута refToMessageID не требуется, поскольку для этой цели может использоваться атрибут InResponseTo элемента SAML <Response>, но, если блок заголовка SOAP <paos:Request> SOAP содержит ID сообщения, то должен использоваться блок заголовка SOAP <paos:Response>.

Как правило, значение блока заголовка <ecp:RelayState> предоставляется по запросу ECP провайдером услуг, но, если провайдер идентификации создает незапрошенный ответ (не получив соответствующего запроса SAML), то он может ввести блок заголовка RelayState, который указывает, на основании взаимного соглашения с провайдером услуг, как обрабатывать последующие взаимодействия с ECP. Это может быть URL ресурса на стороне провайдера услуг.

Если провайдер услуг вводит в свой запрос на ECP блок заголовка SOAP <ecp:RelayState>, или, если провайдер идентификации вводит в свой ответ блок заголовка SOAP <ecp:RelayState, то ECP должен содержать идентичный блок заголовка с ответом SAML, переданным провайдеру услуг. Значение провайдера услуг для этого блока заголовка (если таковое имеется) должно иметь приоритет.

#### 11.4.2.3.8 Провайдер услуг разрешает клиенту доступ или отказывает в доступе

После того как провайдер услуг получил ответ SAML в запросе HTTP (в огибающей SOAP, использующей PAOS), он может ответить, передав служебные данные в ответе HTTP. При использовании этого ответа должны выполняться правила, определенные в профиле SSO браузера в 11.4.1.4.3 и 11.4.1.4.5. То есть при использовании PAOS применяются те же самые правила обработки, которые использовались при получении <Response> со связью HTTP POST.

### 11.4.2.4 Использование схемы профиля ECP

Схема профиля ECP XML определяет блоки заголовка сообщений запрос/ответ SOAP, используемых этим профилем. Далее приводится полный листинг документа этой схемы.

```
<schema
    targetNamespace="urn:oasis:names:tc:SAML:2.0:profiles:SSO:ecp"
    xmlns="http://www.w3.org/2001/XMLSchema"
    xmlns:ecp="urn:oasis:names:tc:SAML:2.0:profiles:SSO:ecp"
    xmlns:samlp="urn:oasis:names:tc:SAML:2.0:protocol"
    xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion"
    xmlns:S="http://schemas.xmlsoap.org/soap/envelope/"
    elementFormDefault="unqualified"
    attributeFormDefault="unqualified"
    blockDefault="substitution"
    version="2.0">
    <import namespace="urn:oasis:names:tc:SAML:2.0:protocol"
        schemaLocation="saml-schema-protocol-2.0.xsd"/>
    <import namespace="urn:oasis:names:tc:SAML:2.0:assertion"
        schemaLocation="saml-schema-assertion-2.0.xsd"/>
    <import namespace="http://schemas.xmlsoap.org/soap/envelope/"
        schemaLocation="http://schemas.xmlsoap.org/soap/envelope/"/>
    <annotation>
        <documentation>
            Document identifier: saml-schema-ecp-2.0
            Location: http://docs.oasis-open.org/security/saml/v2.0/
            Revision history:
              V2.0 (March, 2005):
                Custom schema for ECP profile, first published in SAML 2.0.
        </documentation>
    </annotation>

    <element name="Request" type="ecp:RequestType"/>
    <complexType name="RequestType">
        <sequence>
            <element ref="saml:Issuer"/>
            <element ref="samlp:IDPList" minOccurs="0"/>
        </sequence>
        <attribute ref="S:mustUnderstand" use="required"/>
        <attribute ref="S:actor" use="required"/>
        <attribute name="ProviderName" type="string" use="optional"/>
        <attribute name="IsPassive" type="boolean" use="optional"/>
    </complexType>

    <element name="Response" type="ecp:ResponseType"/>
    <complexType name="ResponseType">
        <attribute ref="S:mustUnderstand" use="required"/>
        <attribute ref="S:actor" use="required"/>
        <attribute name="AssertionConsumerServiceURL" type="anyURI"
use="required"/>
    </complexType>
```

```
    <element name="RelayState" type="ecp:RelayStateType"/>
    <complexType name="RelayStateType">
        <simpleContent>
            <extension base="string">
                <attribute ref="S:mustUnderstand" use="required"/>
                <attribute ref="S:actor" use="required"/>
            </extension>
        </simpleContent>
    </complexType>
</schema>
```

В последующих подразделах описывается, как должны использоваться эти конструкции XML.

#### 11.4.2.4.1 Блок заголовка запроса PAOS: от SP в ECP

Блок заголовка запроса PAOS сообщает об использовании обработки PAOS и содержит следующие атрибуты:

– `responseConsumerURL` [Требуемый]

  Определяет, куда ECP должен передать ответ-ошибку. Используется также для проверки корректности ответа провайдера идентификации, путем перекрестной проверки этого положения на соответствие значению `AssertionServiceConsumerURL` в блоке заголовка ответа ECP. Это значение должно быть тем же, что и `AssertionServiceConsumerURL` (или URL, казанного в метаданных) передаваемое в запросе `<AuthnRequest>`.

  ПРИМЕЧАНИЕ (информативное). – PE22 (см. OASIS PE:2006) предлагает заменить слово `AssertionServiceConsumerURL` в последнем предложении на `AssertionConsumerServiceURL`.

– `service` [Требуемый]

  Указывает, что в этом профиле аутентификации SAML используется служба PAOS. Значение должно быть `urn:oasis:names:tc:SAML:2.0:profiles:SSO:ecp`.

– `SOAP-ENV:mustUnderstand` [Требуемый]

  Значение должно быть `1` (true). Если блок заголовка PAOS не понятен, то должны создаваться, сообщения об ошибках SOAP.

– `SOAP-ENV:actor` [Требуемый]

  Значение должно быть `http://schemas.xmlsoap.org/soap/actor/next`.

– `messageID` [Дополнительный]

  Допускает дополнительную корреляцию ответа. Он может использоваться в этом профиле, но не требуется, поскольку эти функции выполняются протокольным уровнем SAML при помощи атрибута `ID` в `<AuthnRequest>` и атрибута `InResponseTo` в `<Response>`.

Блок заголовка PAOS запроса SOAP не имеет элемента содержания.

#### 11.4.2.4.2 Блок заголовка запроса ECP: от SP в ECP

Блок заголовка SOAP запроса ECP используется для передачи информации, необходимой ECP для выполнения аутентификации запроса. Он обязателен, и его наличие говорит об использовании этого профиля. Он содержит следующие элементы и атрибуты:

– `SOAP-ENV:mustUnderstand` [Требуемый]

  Значение должно быть (true). Если блок заголовка ECP не понятен, то должны создаваться, сообщения об ошибках SOAP.

– `SOAP-ENV:actor` [Требуемый]

  Значение должно быть `http://schemas.xmlsoap.org/soap/actor/next`.

– `ProviderName` [Дополнительный]

  Понятное для человека название запрашивающего провайдера услуг.

– `IsPassive` [Дополнительный]

  Булева величина. Если она = `true`, то провайдер идентификации и клиент сам не должен управлять интерфейсом пользователя с создателем запроса и заметно взаимодействовать с клиентом. Если эта величина не представлена, значение по умолчанию = `true`.

– `<saml:Issuer>` [Требуемый]

  Этот элемент должен содержать уникальный идентификатор запрашивающего провайдера услуг; атрибут `Format` должен быть пропущен или должен иметь значение `urn:oasis:names:tc:SAML:2.0:nameid-format:entity`.

– `<samlp:IDPList>` [Дополнительный]

Дополнительный список провайдеров идентификации, которых признает провайдер услуг и запросы от которых может обрабатывать ECP.

#### 11.4.2.4.3 Блок заголовка сообщения ECP RelayState: от SP в ECP

Блок заголовка ECP RelayState SOAP используется для передачи информации о состоянии от провайдера услуг, которому она понадобится позже при обработке ответа от ECP. Он является дополнительным, но, если он используется, то ECP должен содержать идентичный блок заголовка в ответе этапа 5, показанного на рисунке 11-2. Он содержит следующий атрибуты:

ПРИМЕЧАНИЕ (информативное). – PE27 (см. OASIS PE:2006) предлагает заменить в вышеприведенном тексте этап 5 на этап 7.

– `SOAP-ENV:mustUnderstand` [Требуемый]

Значение должно быть `1` (true). Если блок заголовка ECP не понятен, то должны создаваться, сообщения об ошибках SOAP.

– `SOAP-ENV:actor` [Требуемый]

Значение должно быть `http://schemas.xmlsoap.org/soap/actor/next`.

Содержание элемента блока заголовка – это строка, содержащая информацию о состоянии, созданную запрашивающей стороной. Если это блок представлено, то ECP должен содержать точно такое же значение блока заголовка RelayState в ответе провайдеру услуг на этапе 5. Строчное значение не должно превышать 80 байтов в длину и его целостность должна быть защищена запрашивающей стороной независимо от любых других вариантов защиты, которые могут существовать или не существовать во время передачи сообщения.

Далее приводится пример запроса аутентификации SOAP, передаваемого от провайдера услуг на ECP:

```
<SOAP-ENV:Envelope
 xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion"
 xmlns:samlp="urn:oasis:names:tc:SAML:2.0:protocol"
 xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Header>
    <paos:Request xmlns:paos="urn:liberty:paos:2003-08"
       responseConsumerURL="http://identity-service.example.com/abc"
       messageID="6c3a4f8b9c2d" SOAP-
ENV:actor="http://schemas.xmlsoap.org/soap/actor/next" SOAP-
ENV:mustUnderstand="1"
 service="urn:oasis:names:tc:SAML:2.0:profiles:SSO:ecp">
    </paos:Request>
    <ecp:Request xmlns:ecp="urn:oasis:names:tc:SAML:2.0:profiles:SSO:ecp"
 SOAP-ENV:mustUnderstand="1" SOAP-
ENV:actor="http://schemas.xmlsoap.org/soap/actor/next"
 ProviderName="Service provider X" IsPassive="0">
      <saml:Issuer>https://ServiceProvider.example.com</saml:Issuer>
      <samlp:IDPList>
        <samlp:IDPEntry ProviderID="https://IdentityProvider.example.com"
            Name="Identity Provider X"
            Loc="https://IdentityProvider.example.com/saml2/sso"
        </samlp:IDPEntry>
        <samlp:GetComplete>
        https://ServiceProvider.example.com/idplist?id=604be136-fe91-441e-
afb8
        </samlp:GetComplete>
      </samlp:IDPList>
    </ecp:Request>
    <ecp:RelayState xmlns:ecp="urn:oasis:names:tc:SAML:2.0:profiles:SSO:ecp"
 SOAP-ENV:mustUnderstand="1" SOAP-
ENV:actor="http://schemas.xmlsoap.org/soap/actor/next">
      ...
    </ecp:RelayState>
  </SOAP-ENV:Header>
  <SOAP-ENV:Body>
    <samlp:AuthnRequest> ... </samlp:AuthnRequest>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Как отмечено выше, ECP удаляет блоки заголовка PAOS и ECP из сообщения SOAP до перенаправления запроса аутентификации провайдеру идентификации. Примерный запрос аутентификации от ECP провайдеру идентификации выглядит следующим образом:

```
<SOAP-ENV:Envelope xmlns:SOAP-
ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:samlp="urn:oasis:names:tc:SAML:2.0:protocol"
  <SOAP-ENV:Body>
    <samlp:AuthnRequest> ... </samlp:AuthnRequest>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

#### 11.4.2.4.4 Блок заголовка ответа ECP: от IdP в ECP

Блок заголовка ECP ответа SOAP должен использоваться в ответе от провайдера идентификации, передаваемом на ECP. Он содержит следующий атрибуты:

– `SOAP-ENV:mustUnderstand` [Требуемый]

  Значение должно быть `1` (true). Если блок заголовка ECP не понятен, то должны создаваться, сообщения об ошибках SOAP.

– `SOAP-ENV:actor` [Требуемый]

  Значение должно быть `http://schemas.xmlsoap.org/soap/actor/next`.

– `AssertionConsumerServiceURL` [Требуемый]

  Установлен провайдером идентификации на основе сообщения `<AuthnRequest>` или на основе метаданных провайдера услуг, полученных провайдером идентификации.

ECP должен подтвердить, что это значение соответствует значению ECP, полученному в виде блока `responseConsumerURL` заголовка сообщения PAOS запроса SOAP, которое он получил от провайдера услуг. Поскольку `responseConsumerURL` может быть относительным, а подтверждение `ConsumerServiceURL` является абсолютным, то может потребоваться какая-либо обработка/нормализация.

Этот механизм используется для целей безопасности для подтверждения правильного пункта назначения ответа. Если эти значения не совпадают, то ECP должен ответить провайдеру услуг сообщением SOAP об ошибке и не должен возвращать ответ SAML.

Заголовок ECP в ответе SOAP не имеет элемента содержание.

Далее приведен пример ответа от IdP, предназначенного для ECP.

```
<SOAP-ENV:Envelope
 xmlns:ecp="urn:oasis:names:tc:SAML:2.0:profiles:SSO:ecp"
 xmlns:samlp="urn:oasis:names:tc:SAML:2.0:protocol"
 xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Header>
     <ecp:Response SOAP-ENV:mustUnderstand="1" SOAP-
ENV:actor="http://schemas.xmlsoap.org/soap/actor/next"
AssertionConsumerServiceURL="https://ServiceProvider.example.com/ecp_asserti
on_consumer"/>
  </SOAP-ENV:Header>
  <SOAP-ENV:Body>
    <samlp:Response> ... </samlp:Response>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

#### 11.4.2.4.5 Блок заголовка ответа PAOS: от ECP в SP

Блок заголовка ответа PAOS содержит следующий атрибуты:

– `SOAP-ENV:mustUnderstand` [Требуемый]

  Значение должно быть `1` (true). Если блок заголовка PAOS не понятен, то должны создаваться, сообщения об ошибках SOAP.

– `SOAP-ENV:actor` [Требуемый]

  Значение должно быть `http://schemas.xmlsoap.org/soap/actor/next`.

– `refToMessageID` [Дополнительный]

Допускает корреляцию с запросом PAOS. Этот дополнительный атрибут (и блок заголовка в целом) должен быть добавлен профилем ECP, если соответствующий запрос PAOS определяет атрибут `ID` сообщения. Аналогичные функции реализуются в языке SAML с использованием корреляции сообщений `<AuthnRequest>` и `<Response>`.

Заголовок PAOS в ответе SOAP не имеет элемента содержание.

Далее приведен пример ответа от ECP, предназначенного для SP.

```
<SOAP-ENV:Envelope
 xmlns:paos="urn:liberty:paos:2003-08"
 xmlns:samlp="urn:oasis:names:tc:SAML:2.0:protocol"
 xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Header>
    <paos:Response refToMessageID="6c3a4f8b9c2d" SOAP-
ENV:actor="http://schemas.xmlsoap.org/soap/actor/next/" SOAP-
ENV:mustUnderstand="1"/>
    <ecp:RelayState xmlns:ecp="urn:oasis:names:tc:SAML:2.0:profiles:SSO:ecp"
 SOAP-ENV:mustUnderstand="1" SOAP-
ENV:actor="http://schemas.xmlsoap.org/soap/actor/next">
      ...
    </ecp:RelayState>
  </SOAP-ENV:Header>
  <SOAP-ENV:Body>
    <samlp:Response> ... </samlp:Response>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

#### 11.4.2.5 Аспекты безопасности

Сообщение `<AuthnRequest>` должно быть подписано. В соответствии с правилами, определенными профилем SSO браузера, подтверждения, вложенные в сообщения `<Response>`, должны быть подписаны. Доставка ответа в огибающей SOAP при помощи связи PAOS, по сути, аналогична использованию связи HTTP POST, и используются меры безопасности, пригодные для этой связи.

Целостность заголовков SOAP должна быть защищена, либо при помощи мер по обеспечению безопасности сообщения (Message Security) или путем использования TLS для каждого обмена сообщениям HTTP с клиентом.

Провайдер услуг должен быть аутентифицирован для ECP, например, при помощи аутентификации TLS на стороне сервера.

ECP должен быть аутентифицирован для провайдера идентификации, например в ходе сеанса связи аутентификации. Любой обмен сообщениям HTTP после доставки сообщения `<AuthnRequest>` и до того, как провайдер идентификации вернет сообщение `<Response>`, должен быть безопасно ассоциирован с исходным запросом.

ПРИМЕЧАНИЕ (информативное). – PE20 (см. OASIS PE:2006) предлагает добавить приведенный ниже подраздел для рассмотрения аспектов метаданных ECP:

Здесь также применимы правила, определенные в профиле SSO браузера в разделе 11. В частности, индексированный элемент оконечной точки `<md:AssertionConsumerService>` со связью `urn:oasis:names:tc:SAML:2.0:bindings:PAOS` может использоваться для описания поддерживаемой связи и мест размещения, в которые провайдер идентификации может передавать ответы провайдеру услуг, использующему этот профиль. Более того, оконечная точка `<md:SingleSignOnService>` со связью `urn:oasis:names:tc:SAML:2.0:bindings:SOAP` может использоваться для описания поддерживаемой связи и мест размещения, в которые провайдер услуг может передавать запросы провайдеру идентификации, использующему этот профиль.

### 11.4.3 Профиль обнаружения провайдера идентификации

В настоящем разделе определяются профиль, при помощи которого провайдер услуг может узнать, каких провайдеров идентификации использует клиент с профилем SSO веб-браузера. В вариантах реализации, где имеется несколько провайдеров идентификации, провайдерам услуг требуются средства для определения того, какого (каких) провайдера(ов) идентификации использует клиент. Профиль обнаружения использует маркер HTTP устойчивого состояния, который записан в домене, являющимся в данном варианте реализации общим для провайдеров идентификации и провайдеров услуг. Этот домен, определенный вариантом реализации, в этом профиле называется общим доменом, и маркер HTTP устойчивого состояния, содержащий список провайдеров идентификации, называется маркером HTTP общего домена.

На каких элементах размещены на веб-сервера в общем домене – зависит от варианта реализации, но в данном профиле не рассматривается.

ПРИМЕЧАНИЕ (информативное). – PE32 (см. OASIS PE:2006) предлагает добавить следующий текст для описания требуемой информации:

Идентификация:`urn:oasis:names:tc:SAML:2.0:profiles:SSO:idp-discovery`

Контактная информация: security-services-comment@lists.oasis-open.org

#### 11.4.3.1 Маркер HTTP общего домена

Название этого маркера должно быть следующим "`saml_idp`". Формат значения этого маркера должен представлять собой множество из одного или нескольких значений Унифицированного идентификатора ресурса в кодировке base64, разделенных символом одного пробела. Каждый URI – это уникальный идентификатор провайдера идентификации, как определено в разделе 7. Окончательный набор значений представляется в кодировке URL.

Служба записи маркера HTTP общего домена должна добавить в список уникальный идентификатор провайдера идентификации. Если идентификатор уже имеется в списке, она может удалить и расширить его. Идея состоит в том, чтобы наиболее поздний сеанс связи, установленный с провайдером идентификации, был бы последним в списке.

Этот маркер должен иметь префикс пути передачи = "/". Поле Домен должно быть = ".[common-domain]", где [common-domain] – это общий домен, созданный данным вариантом реализации для использования с этим профилем. Должен иметься также предшествующий период. Этот маркер должен быть отмечен, как безопасный.

Синтаксис маркера HTTP должен соответствовать IETF RFC 2965. Маркер может быть создан для одного сеанса связи или может быть постоянным. Выбор из этих двух вариантов должен быть сделан в рамках варианта реализации, должен быть одинаковым для всех провайдеров идентификации данного варианта реализации.

#### 11.4.3.2 Установка маркера HTTP общего домена

После того как провайдер идентификации аутентифицирует клиента, он может установить маркер HTTP общего домена. Средства, при помощи которых провайдер идентификации устанавливает этот маркер, определяется вариантом реализации при условии, что этот маркер успешно устанавливается с вышеприведенными параметрами. Далее приводится одна из возможных стратегий реализации, ее не следует считать нормативной. Провайдер идентификации может:

- иметь в общем домене ранее установленные для себя псевдоним DNS и псевдоним IP;
- перенаправлять на себя агента пользователя, применяя псевдоним DNS, использующий URL, который определяет "https" как схему URL. Структура URL является секретной для варианта и реализации и может содержать информацию о сеансе связи, необходимую для определения агента пользователя;
- установить маркер HTTP на перенаправленный агент пользователя, используя вышеприведенные параметры;
- перенаправлять обратно на себя агента пользователя или, при необходимости, на провайдера услуг.

#### 11.4.3.3 Получение маркера HTTP общего домена

Когда провайдеру услуг необходимо узнать, каких провайдеров идентификации использует клиент, он инициирует обмен сообщениями, направленный на представление провайдеру услуг маркера HTTP общего домена после того, как это маркер считан сервером HTTP в общем домене.

Если сервер HTTP в общем домене управляется провайдером услуг или если имеются иные структуры, провайдер услуг может использовать сервер HTTP в общем домене для ретрансляции своего сообщения `<AuthnRequest>` провайдеру идентификации для оптимизированного процесса единой регистрации в сети.

Конкретные средства, при помощи которых провайдер услуг считывает маркеры HTTP, определяются вариантом реализации, при условии, что они позволяют агенту пользователя представить маркеры, которые были установлены с параметрами, приведенными в 11.4.3.1. Одна из возможных стратегий реализации может быть описана следующим образом, и ее не следует считать нормативной. Кроме того, она может быть субоптимальной для некоторых приложений:

- иметь в общем домене ранее установленные для себя псевдоним DNS и псевдоним IP;
- перенаправлять на себя агента пользователя, применяя псевдоним DNS, использующий URL, который определяет "https" как схему URL. Структура URL является секретной для варианта и реализации и может содержать информацию о сеансе связи, необходимую для определения агента пользователя;
- перенаправлять обратно на себя агента пользователя или, при необходимости, на провайдера услуг.

### 11.4.4 Профиль единого выхода из системы

После того как клиент аутентифицировал себя для провайдера идентификации, аутентифицирующий элемент может установить сеанс связи с клиентом (как правило, при помощи маркера HTTP, переписывания URL или иных средств, свойственных варианту реализации). Провайдер идентификации может затем передать подтверждения провайдерам услуг или иным доверяющим сторонам, на основе этого события аутентификации; доверяющая сторона может использовать его для установки *своего собственного* сеанса связи с клиентом.

В такой ситуации, провайдер идентификации может действовать как ответственный орган сеанса связи, а доверяющие стороны – как участники сеанса связи. Немного позже, клиент может принять решение о завершении своего сеанса связи либо с отдельным участником сеанса связи, либо со всеми участниками сеанса связи в данном сеансе связи, регулируемом ответственным органом сеанса связи. Первый случай не входит в область рассмотрения настоящей Рекомендации. Последний же случай, может быть выполнен, используя этот профиль протокола единого выхода из системы SAML (см. 11.4).

Клиент (или администратор, завершающий сеанс связи клиента) может принять решение о завершении этого "глобального" сеанса связи, обратившись либо в ответственный орган сеанса связи, либо к отдельному участнику сеанса связи. Кроме того, провайдер идентификации, действующий в качестве ответственного органа сеанса связи, может *сам* действовать как участник сеанса связи в ситуациях, когда он является доверяющей стороной для подтверждения другого провайдера идентификации в отношении данного клиента.

Профиль позволяет комбинировать протокол с синхронной связью, например, связью SOAP, или с асинхронными связями "прямого канала", такими связями как HTTP Redirect, HTTP POST или HTTP Artifact. Связь прямого канала может потребоваться, например, в случаях, когда сеанс связи клиента выходит только на агента пользователя в форме маркера HTTP, и требуется прямое взаимодействие между агентом пользователя и участником сеанса связи или ответственным органом сеанса связи. Как будет рассмотрено ниже, участники сеанса связи должны, по возможности, использовать связь "прямого канала", когда инициируют этот профиль, для максимизации вероятности того, что ответственный орган сеанса связи сможет успешно распространить действие по единому выходу из системы на всех участников.

#### 11.4.4.1 Требуемая информация

**Идентификация**: `urn:oasis:names:tc:SAML:2.0:profiles:SSO:logout`

**Контактная информация**: security-services-comment@lists.oasis-open.org

**Описание**: Приводится ниже.

**Обновления**: Нет

#### 11.4.4.2 Обзор профиля

На рисунке 11-3 показан базовый формат выполнения единого выхода из системы.

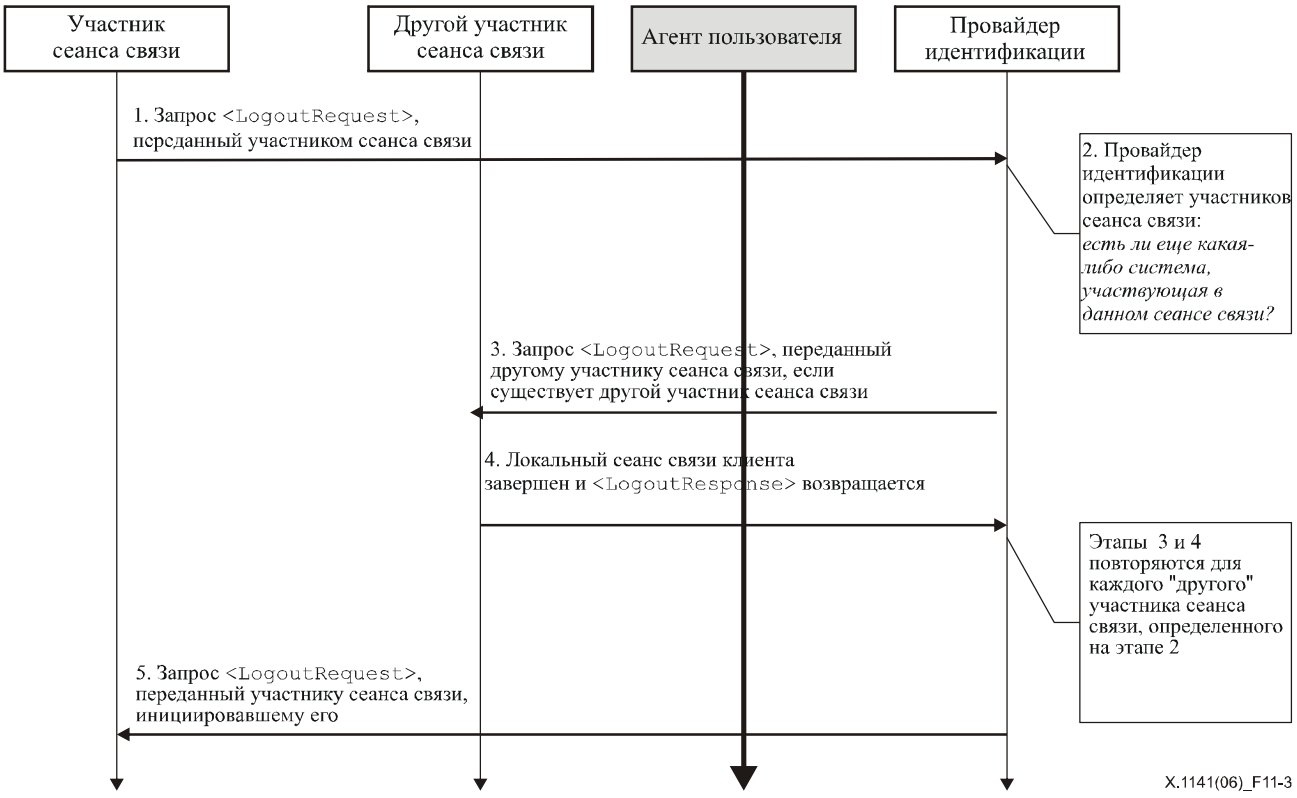


**Рисунок 11-3/X.1141 – формат выполнения единого выхода из системы**

Серый прямоугольник "агент пользователя" показывает, что обмен сообщениями может проходить через агента пользователя, или может быть прямым обменом между элементами системы, в зависимости от используемой связи SAML для реализации профиля.

Профилем описываются следующие этапы. В рамках отдельного этапа может быть организован один или несколько обменов сообщениями в зависимости от связи, используемой для этого этапа и функций, определяемых вариантом реализации.

1) **Запрос<LogoutRequest>, переданный участником сеанса связи провайдеру идентификации**

   На этапе 1 участник сеанса связи инициирует единый выход из системы и завершает сеанс(ы) связи клиента, передавая сообщение `<LogoutRequest>` провайдеру идентификации, о которого он получил соответствующее подтверждение аутентификации. Этот запрос может быть передан провайдеру идентификации непосредственно или опосредованно через агент пользователя.

2) **Провайдер идентификации определяет участников сеанса связи**

На этапе 2 провайдер идентификации использует содержание сообщения `<LogoutRequest>` (или, если он сам инициирует единый выход из системы, то какие-либо иные механизмы) для определения завершаемого(ых) сеанса(ов) связи. Если других участников сеанса связи нет, то профиль выполняется с этапа 5. В противном случае для каждого определенного участника сеанса связи повторяются этапы 3 и 4.

3) **Запрос <LogoutRequest>, переданный провайдером идентификации участнику сеанса связи/ответственному органу**

На этапе 3 провайдер идентификации передает сообщение `<LogoutRequest>` участнику сеанса связи или ответственному органу сеанса связи, связанным с одним или несколькими завершаемыми сеансами. Этот запрос может быть передан непосредственно на элемент, или опосредованно через агента пользователя (если он соответствует форме запроса, полученного на этапе 1).

4) **Участник сеанса связи/ответственный орган передает <LogoutResponse> провайдеру идентификации**

На этапе 4 участник сеанса связи или ответственный орган сеанса связи завершает сеанс(ы) связи клиента, как указано в запросе (если возможно) и возвращает `<LogoutResponse>` провайдеру идентификации. Этот ответ может быть возвращен провайдеру идентификации непосредственно или опосредованно через агент пользователя (если он соответствует форме запроса, полученного на этапе 3).

5) **Провайдер идентификации передает <LogoutResponse> участнику сеанса связи**

На этапе 5 провайдер идентификации передает сообщение `<LogoutResponse>` исходному запрашивающему участнику сеанса связи. Этот ответ может быть возвращен участнику сеанса связи непосредственно или опосредованно через агент пользователя (если он соответствует форме запроса, полученного на этапе 1).

Провайдер идентификации (действуя как ответственный орган сеанса связи) может инициировать выполнение этого профиля на этапе 2 и передать `<LogoutRequest>` всем участникам сеанса связи, также пропуская этап 5.

#### 11.4.4.3 Описание профиля

Если профиль инициируется участником сеанса связи, начинаем с 11.4.4.3.1. Если он инициируется провайдером идентификации, начинаем с 11.4.4.3.2. В нижеприведенном описании используется обозначение:

– **Служба Единого выхода из системы**

Это оконечная точка протокола запроса аутентификации на стороне провайдера идентификации или участника сеанса связи, на которую доставляются сообщения `<LogoutRequest>` или `<LogoutResponse>` (или представляющие их артефакты). Для запросов и ответов может использоваться одна и та же или разные оконечные очки.

##### 11.4.4.3.1 <LogoutRequest>, переданный участником сеанса связи провайдеру идентификации

Если профиль выхода из системы инициируется участником сеанса связи, он проверяет подтверждение(я) аутентификации, которые он получает, относительно завершаемого(ых) сеанса(ов) связи, и собирает значение(я) `SessionIndex`, которые он получает от провайдера идентификации. Если в обмене участвует несколько провайдеров идентификации, то профиль должен независимо повторяться для каждого из них.

Для того чтобы инициировать профиль, участник сеанса связи передает на оконечную точку службы единого выхода из системы провайдера идентификации сообщение `<LogoutRequest>`, содержащее один или несколько применимых элементов `<SessionIndex>`. Как минимум один элемент должен быть включен. Для определения местоположения этой оконечной точки и связей, поддерживаемых провайдером идентификации, могут использоваться метаданные.

**Асинхронные связи (прямой канал)**

Участник сеанса связи должен (если представлен агент пользователя клиента) использовать асинхронную связь, такую как HTTP Redirect, POST или Артефакт (см. раздел 10), для передачи запроса провайдеру идентификации через агента пользователя. Провайдер идентификации должен затем распространить все требуемые сообщения о выходе из системы дополнительным участникам сеанса связи, как требуется синхронной или асинхронной связью. Использование асинхронной связи для исходного запроса предпочтительно потому, что это дает провайдеру идентификации наилучший шанс для успешного распространения сообщений о выходе из системы остальным участникам сеанса связи во время выполнения этапа 3 в 11.4.4.2.

Если используется связь HTTP Redirect или POST, то сообщение `<LogoutRequest>` доставляется провайдеру идентификации на этом этапе. Если используется связь HTTP Artifact, то профиль разрешения артефакта, определенный в 11.4.6, используется провайдером идентификации, который выполняет обратный вызов участника сеанса связи для получения от него сообщения `<LogoutRequest>`, используя, например, связь SOAP.

Рекомендуется, чтобы обмены сообщениями HTTP на этом этапе выполнялись по TLS 1.0 для сохранения конфиденциальности и целостности сообщения. Если используется связь HTTP Redirect или POST, то сообщение `<LogoutRequest>` должно быть подписано. Связь HTTP Artifact, если она используется, также предоставляет альтернативные средства аутентификации создателя запроса, когда указывается артефакт.

Каждая из связей предоставляет механизм RelayState, который участник сеанса связи может использовать для связи обмена сообщений профиля с исходным запросом. Участник сеанса связи должен минимально возможно открывать информацию в значении `RelayState`, если только для использования профиля не указаны требования в таких мерах секретности.

**Синхронные связи (Обратный канал)**

Альтернативно, участник сеанса связи может использовать синхронную связь, такую как связь SOAP (см. раздел 10), для передачи запроса непосредственно провайдеру идентификации. Провайдер идентификации должен затем распространить все требуемые сообщения о выходе из системы дополнительным участникам сеанса связи, как это требуется, используя синхронную связь. Запрашивающая сторона должна аутентифицировать себя для провайдера идентификации, либо подписав `<LogoutRequest>`, либо используя любые другие механизмы, поддерживаемые связью.

Определяемые профилем правила для содержания сообщения `<LogoutRequest>` приведены в 11.4.4.4.1.

#### 11.4.4.3.2 Провайдер идентификации определяет участников сеанса связи

Если профиль выхода из системы инициируется провайдером идентификации, или если, получив достоверный запрос `<LogoutRequest>`, провайдер идентификации его обрабатывает, он должен проверить идентификатор и элементы `<SessionIndex>`, определяя те сеансы связи, которые должны быть завершены.

Провайдер идентификации затем выполняет этапы 3 и 4 на рисунке 11-3 для каждого элемента, участвующего в завершаемых сеансах связи, отличных от исходного запрашивающего участника сеанса связи (если таковые имеются), как описано в 8.2.7.

#### 11.4.4.3.3 Запрос <LogoutRequest>, передаваемый провайдером идентификации участнику сеанса связи/ответственному органу

Для распространения сообщения о выходе из системы провайдер идентификации передает свое собственное сообщение `<LogoutRequest>` ответственному органу или участнику завершаемого сеанса связи. Этот запрос передается с использованием связи SAML, соответствующей возможностям отвечающей стороны и доступности агента пользователя на стороне провайдер идентификации.

Как правило, связь, при помощи которой был получен исходный запрос на этапе 1 на рисунке 11-3, не указывает связь, которая может использоваться на этом этапе, за исключением той, что уже была указана на этапе 1, используя синхронную связь, которая обходит ограничения, накладываемые агентом пользователя на провайдера идентификации для использования аналогичной связи для распространения дополнительных запросов.

Определяемые профилем правила для содержания сообщения `<LogoutRequest>` приведены в 11.4.4.4.1.

#### 11.4.4.3.4 Участник сеанса связи/ответственный орган передает <LogoutResponse> провайдеру идентификации

Участник сеанса связи/ответственный орган должен обработать сообщение `<LogoutRequest>`, как определено в 8.2.7. После обработки сообщения или при появлении ошибки этот элемент должен передать запрашивающему провайдеру идентификации сообщение `<LogoutResponse>`, содержащее соответствующий код статуса, с целью завершения обмена по протоколу SAML.

**Синхронные связи (Обратный канал)**

Если провайдер идентификации использует синхронную связь, такую как SOAP (см. раздел 10), то ответ возвращается непосредственно для завершения синхронного соединения. Отвечающая сторона должна аутентифицировать себя для запрашивающего провайдера идентификации, либо подписав `<LogoutRequest>`, либо используя любые другие механизмы, поддерживаемые связью.

**Асинхронные связи (Прямой канал)**

Если провайдер идентификации использует асинхронную связь, такую как HTTP Redirect, POST или Артефакт (см. раздел 10), то сообщение `<LogoutResponse>` (или артефакт) возвращается через агента пользователя на оконечную точку службы единого выхода из системы провайдера идентификации. Для определения местоположения этой оконечной точки и связей, поддерживаемых провайдером идентификации, могут использоваться метаданные. Может использоваться любая асинхронная связь, поддерживаемая обоими элементами.

Если используется связь HTTP Redirect или POST, то сообщение `<LogoutRequest>` доставляется провайдеру идентификации на этом этапе. Если используется связь HTTP Artifact, то профиль разрешения артефакта, определенный в 11.4.6, используется провайдером идентификации, который выполняет обратный вызов участника сеанса связи для получения от него сообщения `<LogoutRequest>`, используя, например, связь SOAP.

Рекомендуется, чтобы обмены сообщениями HTTP на этом этапе выполнялись по TLS 1.0 для сохранения конфиденциальности и целостности сообщения. Если используется связь HTTP Redirect или POST, то сообщение `<LogoutRequest>` должно быть подписано. Связь HTTP Artifact, если она используется, также предоставляет альтернативные средства аутентификации создателя запроса, когда указывается артефакт.

Определяемые профилем правила для содержания сообщения `<LogoutResponse>` приведены в 11.4.4.4.2.

#### 11.4.4.3.5 Провайдер идентификации передает <LogoutResponse> участнику сеанса связи

После обработки запроса `<LogoutRequest>` исходного участника сеанса связи, как описано в предыдущих этапах, провайдер идентификации должен ответить на исходный запрос, передав ответ `<LogoutResponse>`, содержащий соответствующий код состояния для завершения обмена по протоколу SAML.

Ответ передается исходному участнику сеанса связи, используя связь SAML, соответствующую связи, использованной в исходном запросе, возможностям отвечающей стороны и доступности агента пользователя на стороне провайдера идентификации. Если на этапе 1 (рисунок 11-3) была использована асинхронная связь, то может использоваться любая связь, поддерживаемая обоими элементами.

Определяемые профилем правила для содержания сообщения <LogoutResponse> приведены в 11.4.4.4.2

### 11.4.4.4 Использование протокола Единого выхода из системы

В настоящем разделе описывается использование сообщений `<LogoutRequest>` и `<LogoutResponse>`.

#### 11.4.4.4.1 Использование сообщения <LogoutRequest>

Элемент `<Issuer>` должен быть представлен и должен содержать уникальный идентификатор запрашивающего элемента; атрибут `Format` должен быть пропущен или должен иметь значение `urn:oasis:names:tc:SAML:2.0:nameid-format:entity`.

Запрашивающая сторона должна аутентифицировать себя для отвечающей стороны и гарантировать целостность сообщения, подписав сообщение, либо используя любые другие механизмы, поддерживаемые связью.

Клиент должен быть идентифицирован в запросе с использованием идентификатора, который **точно соответствует** идентификатору в подтверждении аутентификации, которое запрашивающая сторона передавала или принимала в отношении завершаемого сеанса связи согласно правилам соответствия, определенным в 8.2.7.

Если запрашивающая сторона является участником сеанса связи, то в запросе должен содержаться как минимум один элемент `<SessionIndex>`. Если запрашивающая сторона является ответственным органом сеанса связи (или действует от его имени), то она может пропустить такие элементы для указания того, что завершаются все сеансы связи данного клиента.

ПРИМЕЧАНИЕ (информативное). – PE38 (см. OASIS PE:2006) разъясняет вышеприведенный параграф следующим образом:

Если запрашивающая сторона является участником сеанса связи, то в запросе должен содержаться как минимум один элемент `<SessionIndex>`. (На основании 11.4 участник сеанса связи всегда принимает атрибут сеанса связи `Index` в элементах `<saml:AuthnStatement>`, которые он принимает для инициации сеанса связи.) Если запрашивающая сторона является ответственным органом сеанса связи (или действует от его имени), то она может пропустить такие элементы для указания того, что завершаются все сеансы связи данного клиента.

#### 11.4.4.4.2 Использование сообщения <LogoutResponse>

Элемент `<Issuer>` должен быть представлен и должен содержать уникальный идентификатор отвечающего элемента; атрибут `Format` должен быть пропущен или должен иметь значение `urn:oasis:names:tc:SAML:2.0:nameid-format:entity`.

Отвечающая сторона должна аутентифицировать себя для запрашивающей стороны гарантировать целостность сообщения, подписав сообщение, либо используя любые другие механизмы, поддерживаемые связью.

### 11.4.4.5 Использование метаданных

Элемент оконечной точки `<md:SingleLogoutService>` описывает поддерживаемые связи и место(а) размещения, куда элемент может передавать запросы и ответы, используя этот профиль. Запрашивающая сторона, если она шифрует идентификатор клиента, может использовать элемент `<md:KeyDescriptor>` отвечающей стороны с атрибутом `use`, имеющим значение `encryption` для определения соответствующего алгоритма шифрования и установок для использования, вместе с открытым ключом, который должен использоваться при доставке ключей шифрования.

## 11.4.5 Профиль управления идентификатором имени

В сценарии, поддерживаемом профилем управления идентификатором имени, провайдер идентификации обменивается с провайдером услуг данными о существующем идентификаторе клиента, что позволяет им использовать в течение некоторого времени общий идентификатор. Следовательно, провайдер идентификации может пожелать уведомить провайдера услуг об изменении формата и/или значении, которые он будет использовать для определения того же клиента в будущем. И, наоборот, провайдер услуг может пожелать прикрепить к клиенту свой собственный "псевдоним" для того, чтобы быть уверенным в том, что провайдер идентификации будет включать его в сообщения в будущих обменах данными об этом клиенте. И, наконец, один из провайдеров может пожелать сообщить другому о том, что он более не передает и не принимает сообщения, в которых используется определенный идентификатор. Для того чтобы реализовать эти сценарии, используется профиль Протокола SAML управления идентификатором имени.

ПРИМЕЧАНИЕ (информативное). – PE12 (см. OASIS PE:2006) предлагает переписать второе предложение в вышеприведенном параграфе следующим образом:

Следовательно, провайдер идентификации может пожелать уведомить провайдера услуг об изменении значении, которое он будет использовать для определения того же клиента в будущем.

Профиль позволяет комбинировать протокол с синхронной связью, такой как SOAP, или с асинхронной связью "прямой канал", такой как HTTP Redirect, POST или Артефакт. Может потребоваться связь прямого канала, например, в том случае, когда требуется прямое взаимодействие между агентом пользователя и отвечающим провайдером для выполнения обмена.

#### 11.4.5.1 Требуемая информация

**Идентификация**: `urn:oasis:names:tc:SAML:2.0:profiles:SSO:nameid-mgmt`

**Контактная информация**: security-services-comment@lists.oasis-open.org

**Описание**: Приводится ниже.

**Обновления**: Нет.

#### 11.4.5.2 Обзор профиля

На рисунке 11-4 показан базовый формат для профиля управления идентификатором имени.

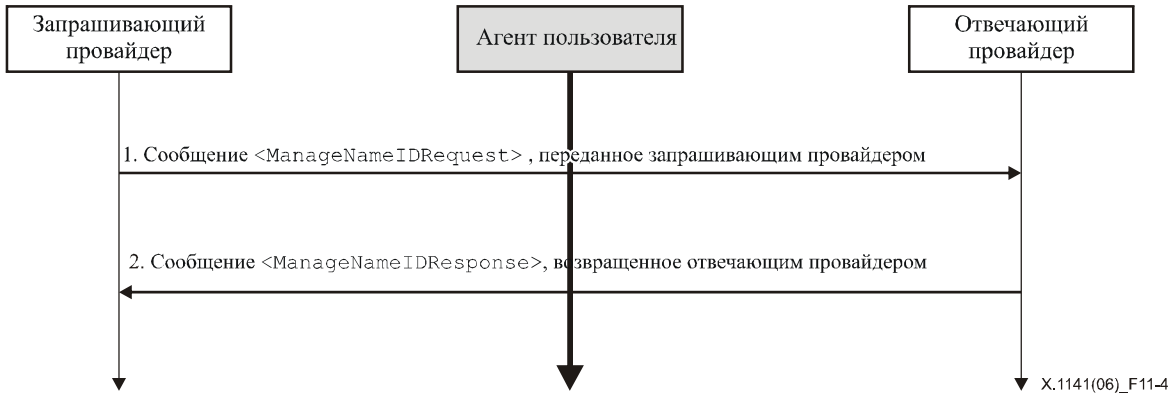


**Рисунок 11-4/X.1141 – Профиль управления идентификатором имени**

Серый прямоугольник "агент пользователя" показывает, что обмен сообщениями может проходить через агента пользователя, или может быть прямым обменом между элементами системы, в зависимости от используемой связи SAML для реализации профиля.

Профилем описываются следующие этапы. В рамках отдельного этапа может быть организован один или несколько обменов сообщениями в зависимости от связи, используемой для этого этапа и функций, определяемых вариантом реализации.

1) **Запрос <ManageNameIDRequest> переданный провайдером идентификации/услуг**

   На этапе 1 провайдер идентификации или провайдер услуг инициирует выполнение профиля, передавая сообщение `<ManageNameIDRequest>` другому провайдеру, которого он желает проинформировать об изменении. Этот запрос может быть передан непосредственно отвечающему провайдеру или опосредованно через агента пользователя.

2) **Ответ <ManageNameIDResponse> переданный отвечающим провайдером идентификации/услуг**

   На этапе 2 отвечающий провайдер (после обработки запроса) передает запрашивающему провайдеру сообщение `<ManageNameIDResponse>`. Этот ответ может быть возвращен непосредственно принимающему отвечающему провайдеру или опосредованно через агента пользователя (если он соответствует форме запроса на этапе 1).

#### 11.4.5.3 Описание профиля

В нижеприведенных описаниях используются следующие обозначения:

**Служба управления идентификатором имени**

Это оконечная точка протокола управления идентификатором имени на стороне провайдера идентификации или провайдера услуг, на которую доставляются сообщения `<ManageNameIDRequest>` или `<ManageNameIDResponse>` (или представляющий их артефакт). Для запросов и ответов может использоваться одна и та же или разные оконечные очки.

##### 11.4.5.3.1 <ManageNameIDRequest>, переданный запрашивающим провайдером идентификации/услуг

Для того чтобы инициировать выполнение профиля, принимающий провайдер передает на оконечную точку службы управления идентификатором имени другого провайдера сообщение `<ManageNameIDRequest>`. Для определения местоположения этой оконечной точки и связей, поддерживаемых отвечающим провайдером, могут использоваться метаданные.

– **Синхронные связи (Обратный канал)**

  Принимающий провайдер может использовать синхронную связь, такую как SOAP (см. раздел 10), для передачи запроса непосредственно другому провайдеру. Запрашивающая сторона должна аутентифицировать для другого провайдера, либо подписав `<ManageNameIDRequest>`, либо используя любые другие механизмы, поддерживаемые связью.

– **Асинхронные связи (Прямой канал)**

  Альтернативно, принимающий провайдер может (если представлен агент пользователя клиента) использовать асинхронную связь, такую как HTTP Redirect, POST или Артефакт (см. раздел 10) для передачи запроса другому провайдеру через агента пользователя.

Если используется связь HTTP Redirect или POST, то сообщение `<ManageNameIDRequest>` доставляется другому провайдеру на этом этапе. Если используется связь HTTP Artifact, профиль разрешения артефакта, определенный в 11.4.6, используется другим провайдером, который выполняет обратный вызов принимающего провайдера для получения от него сообщения `<ManageNameIDRequest>`, используя, например, связь SOAP.

Рекомендуется, чтобы обмены сообщениями HTTP на этом этапе выполнялись по TLS 1.0 для сохранения конфиденциальности и целостности сообщения. Если используется связь HTTP Redirect или POST, то сообщение `<ManageNameIDRequest>` должно быть подписано. Связь HTTP Artifact, если она используется, также предоставляет альтернативные средства аутентификации создателя запроса, когда указывается артефакт.

Каждая из этих связей предоставляет механизм RelayStat, который может использовать принимающий провайдер для связи профиля обмена с исходным запросом. Принимающий провайдер должен минимально возможно открывать информацию в значении RelayState, если только для использования профиля не указаны требования в таких мерах секретности.

Определяемые профилем правила для содержания сообщения `<ManageNameIDRequest>` перечислены в 11.4.5.4.1.

#### 11.4.5.3.2 <ManageNameIDResponse>, переданный отвечающим провайдером идентификации/услуг

Получатель должен обработать сообщение `<ManageNameIDRequest>`. После того как сообщение обработано или в случае появления ошибки, получатель должен передать принимающему провайдеру сообщение `<ManageNameIDResponse>`, содержащее соответствующий код состояния, для завершения обмена по протоколу SAML.

– **Синхронные связи (Обратный канал)**

  Если принимающий провайдер использовал синхронную связь, такую как SOAP (см. раздел 10), ответ возвращается непосредственно ему для завершения синхронной связи. Отвечающая сторона должна аутентифицировать себя для принимающего провайдера, либо подписав <ManageNameIDResponse>, либо используя любые другие механизмы, поддерживаемые связью.

– **Асинхронные связи (Прямой канал)**

  Если принимающий провайдер использует асинхронную связь, такую как HTTP Redirect, HTTP POST или HTTP Artifact (см. раздел 10), то сообщение `<ManageNameIDResponse>` (или артефакт) возвращается через агента пользователя на оконечную точку службы управления идентификатором имени принимающего провайдера. Для определения местоположения этой оконечной точки и связей, поддерживаемых принимающим провайдером, могут использоваться метаданные. Может использоваться любая связь, поддерживаемая обоими элементами.

Если используется связь HTTP Redirect или POST, то сообщение `<ManageNameIDResponse>` доставляется принимающему провайдеру на этом этапе. Если используется связь HTTP Artifact, то профиль разрешения артефакта, определенный в 11.4.6, используется принимающим провайдером, который выполняет обратный вызов отвечающего провайдера для получения от него сообщения `<ManageNameIDResponse>`, используя, например, связь SOAP.

Рекомендуется, чтобы обмены сообщениями HTTP на этом этапе выполнялись по TLS 1.0 для сохранения конфиденциальности и целостности сообщения. Если используется связь HTTP Redirect или POST, то сообщение `<ManageNameIDResponse>` должно быть подписано. Связь HTTP Artifact, если она используется, также предоставляет альтернативные средства аутентификации создателя ответа, когда указывается артефакт.

Определяемые профилем правила для содержания сообщения `<ManageNameIDResponse>` перечислены в 11.4.5.4.2.

### 11.4.5.4 Использование протокола управления идентификатором имени

В настоящем разделе рассматривается использование сообщений `ManageNameIDRequest` и `ManageNameIDResponse`.

#### 11.4.5.4.1 Использование сообщения <ManageNameIDRequest>

Элемент `<Issuer>` должен быть представлен и должен содержать уникальный идентификатор запрашивающего элемента; атрибут `Format` должен быть пропущен или должен иметь значение `urn:oasis:names:tc:SAML:2.0:nameid-format:entity`.

Запрашивающая сторона должна аутентифицировать себя для отвечающей стороны и гарантировать целостность сообщения, подписав сообщение, либо используя любые другие механизмы, поддерживаемые связью.

#### 11.4.5.4.2 Использование сообщения <ManageNameIDResponse>

Элемент `<Issuer>` должен быть представлен и должен содержать уникальный идентификатор отвечающего элемента; атрибут `Format` должен быть пропущен или должен иметь значение `urn:oasis:names:tc:SAML:2.0:nameid-format:entity`.

Отвечающая сторона должна аутентифицировать себя для запрашивающей стороны и гарантировать целостность сообщения, подписав сообщение, либо используя любые другие механизмы, поддерживаемые связью.

### 11.4.5.5 Использование метаданных

Элемент оконечной точки `<md:ManageNameIDService>` описывает поддерживаемые связи и место(а) размещения, куда элемент может передавать запросы и ответы, используя этот профиль. Запрашивающая сторона, если она шифрует идентификатор клиента, может использовать элемент `<md:KeyDescriptor>` отвечающей стороны с атрибутом `use`, имеющим значение `encryption` для определения соответствующего алгоритма шифрования и установок, которые должны использоваться, вместе с открытым ключом, который должен использоваться при доставке ключей шифрования.

## 11.4.6 Профиль разрешения артефакта

В разделе 10 определяется протокол разрешения артефакта артефакт для введении артефакта SAML в соответствующее сообщение протокола. Связь HTTP Artifact (см. раздел 10) усиливает работу этого механизма, передавая протокольные сообщения SAML посредством ссылок. Этот профиль описывает использование этого протокола с синхронной связью, такой как связь SOAP, определенная в разделе 10.

### 11.4.6.1 Требуемая информация

**Идентификация**: `urn:oasis:names:tc:SAML:2.0:profiles:artifact`

**Контактная информация**: security-services-comment@lists.oasis-open.org

**Описание**: Приводится ниже.

**Обновления**: Нет.

### 11.4.6.2 Обзор профиля

Правила обмена сообщениями и базовые правила обработки, которые управляют работой этого профиля, довольно широко представлены в разделе 8, где определены сообщения, которые должны передаваться в комбинации со связью, используемой для передачи сообщений. В разделе 10 определяется связь для обмена сообщениями как SOAP V1.1. Если в настоящей Рекомендации специально не указано иного, действуют все требования, приведенные в данной спецификации.

На рисунке 11-5 показан базовый формат для профиля разрешения артефакта.

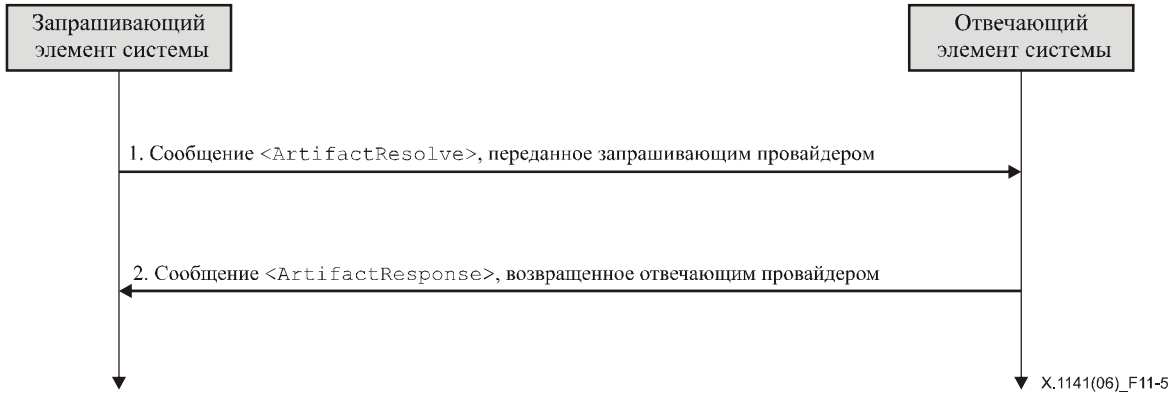


**Рисунок 11-5/X.1141 – Базовый формат для профиля разрешения артефакта**

Профилем описываются следующие этапы.

1) **Сообщение <ArtifactResolve>, переданное запрашивающим элементом**

   На этапе 1 запрашивающая сторона инициирует выполнение профиля, передавая сообщение `<ArtifactResolve>` создателю артефакта.

2) **Сообщение <ArtifactResponse>, переданное отвечающим элементом**

   На этапе 2 отвечающая сторона (после обработки запроса) передает сообщение запрашивающей стороне `<ArtifactResponse>`.

#### 11.4.6.3 Описание профиля

В нижеприведенных описаниях используются следующие обозначения:

– **Служба разрешения артефакта**

  Это оконечная точка протокола разрешения артефакта на стороне создателя артефакта, на которую доставляются сообщения `<ArtifactResolve>`.

##### 11.4.6.3.1 Сообщение <ArtifactResolve>, переданное запрашивающим элементом

Для того чтобы инициировать выполнение профиля, запрашивающая сторона, получив артефакт и определив его создателя, используя атрибут `SourceID`, передает на оконечную точку службы разрешения артефакта создателя артефакта сообщение `<ArtifactResolve>`, содержащее этот артефакт. Для определения местоположения этой оконечной точки и связей, поддерживаемых создателем артефакта, могут использоваться метаданные.

Запрашивающая сторона должна использовать синхронную связь, такую как SOAP (см. раздел 10), для передачи запроса непосредственно создателю артефакта. Запрашивающая сторона должна аутентифицировать себя для отвечающей стороны, либо, подписав сообщение `<ArtifactResolve>`, либо используя любые другие механизмы, поддерживаемые связью. Конкретные профили, которые используют связь HTTP Artifact, могут накладывать дополнительные требования, так что эта аутентификация является обязательной.

Определяемые профилем правила для содержания сообщения `<ArtifactResolve>` перечислены в 11.4.6.4.1.

##### 11.4.6.3.2 Сообщение <ArtifactResponse>, переданное отвечающим элементом

Создатель артефакта должен обработать сообщение `<ArtifactResolve>`, как определено в разделе 8. После того как сообщение обработано, или в случае появления ошибки, создатель артефакта должен возвратить запрашивающей стороне сообщение `<ArtifactResponse>`, содержащее соответствующий код состояния для завершения обмена по протоколу SAML. Если обработка выполнена успешно, то в него будет включен также соответствующий артефакт, указывающий протокольное сообщение SAML.

Отвечающая сторона должна аутентифицировать себя для запрашивающей стороны, подписав `<ArtifactResponse>` либо используя любые другие механизмы, поддерживаемые связью.

Определяемые профилем правила для содержания сообщения `<ArtifactResponse>` перечислены в 11.4.6.4.2.

#### 11.4.6.4 Использование протокола разрешения артефакта

В настоящем разделе описывается использование сообщений `ArtifactResolve` и `ArtifactResponse`.

##### 11.4.6.4.1 Использование сообщения <ArtifactResolve>

Элемент `<Issuer>` должен быть представлен и должен содержать уникальный идентификатор запрашивающего элемента; атрибут `Format` должен быть пропущен или должен иметь значение `urn:oasis:names:tc:SAML:2.0:nameid-format:entity`.

Запрашивающая сторона должна аутентифицировать себя для отвечающей стороны и гарантировать целостность сообщения, подписав сообщение либо используя любые другие механизмы, поддерживаемые связью. Конкретные профили, которые используют связь HTTP Artifact, могут накладывать дополнительные требования, так что эта аутентификация является обязательной.

##### 11.4.6.4.2 Использование сообщения <ArtifactResponse>

Элемент `<Issuer>` должен быть представлен и должен содержать уникальный идентификатор создателя артефакта; атрибут `Format` должен быть пропущен или должен иметь значение `urn:oasis:names:tc:SAML:2.0:nameid-format:entity`.

Отвечающая сторона должен аутентифицировать себя для запрашивающей стороны и гарантировать целостность сообщения, подписав сообщение, либо используя любые другие механизмы, поддерживаемые связью.

#### 11.4.6.5 Использование метаданных

В разделе 9 определяется индексированный элемент оконечной точки `<md:ArtifactResolutionService>`, который описывает поддерживаемые связи и место(а) размещения, куда запрашивающая сторона может передавать запросы, используя этот профиль. Атрибут `index` используется для того, чтобы отличать друг от друга возможные оконечные точки, которые могут быть указаны посредством ссылок в поле `EndpointIndex` артефакта.

### 11.4.7 Профиль запроса/изучения подтверждения

В разделе 10 определяется протокол для запроса существующих подтверждений посредством ссылки или изучения базового объекта и дополнительных критериев, определяемых утверждением. Этот профиль описывает использование этого протокола с синхронной связью, такой как SOAP, определенной в разделе 10.

#### 11.4.7.1 Требуемая информация

**Идентификация**: `urn:oasis:names:tc:SAML:2.0:profiles:query`

**Контактная информация**: security-services-comment@lists.oasis-open.org

**Описание**: Приводится ниже.

**Обновления**: Нет.

#### 11.4.7.2 Обзор профиля

Правила обмена сообщениями и базовые правила обработки, которые управляют работой этого профиля, довольно широко представлены в разделе 8, где определены сообщения, которые должны передаваться в комбинации со связью, используемой для передачи сообщений. В разделе 10 определяется связь для обмена сообщениями как SOAP V1.1. Если в настоящей Рекомендации специально не указано иного, действуют все требования, приведенные в данной спецификации.

На рисунке 11-6 показан базовый формат для профиля запроса/изучения.

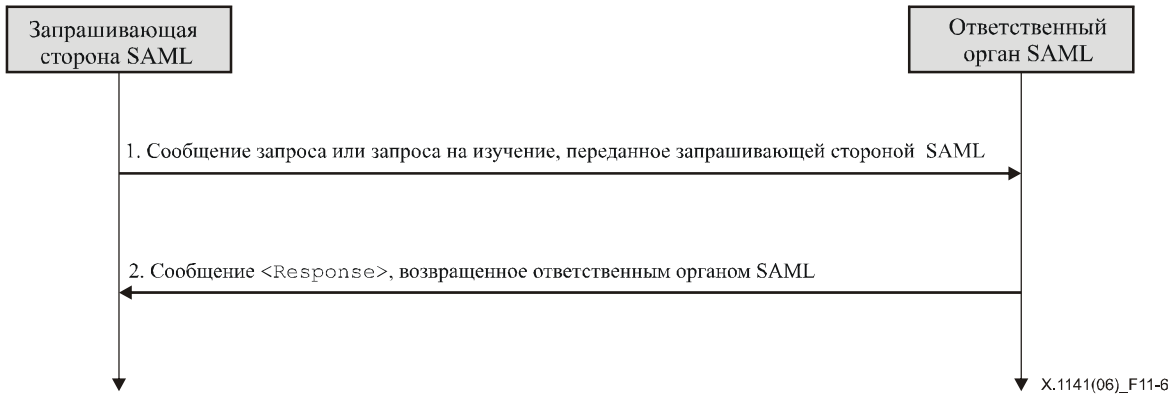


**Рисунок 11-6/X.1141 – Базовый формат для профиля запроса/изучения**

Профилем описываются следующие этапы:

1) **Запрос, переданный запрашивающей стороной SAML**

   На этапе 1 запрашивающая сторона SAML инициирует выполнение профиля, передавая ответственному органу SAML сообщение `<AssertionIDRequest>`, `<SubjectQuery>`, `<AuthnQuery>`, `<AttributeQuery>` или `<AuthzDecisionQuery>`.

2) **<Response>, переданный ответственным органом SAML**

   На этапе 2 отвечающий ответственный орган SAML (после обработки запроса) передает запрашивающей стороне SAML сообщение `<Response>`.

#### 11.4.7.3 Описание профиля

В нижеприведенных описаниях используются следующие обозначения:

– **Служба запроса/изучения**

  Это оконечная точка протокола запроса/изучения на стороне ответственного органа SAML, на которую доставляются запросы на изучение или сообщения `<AssertionIDRequest>`.

##### 11.4.7.3.1 Запрос/запрос на изучение, переданный запрашивающей стороной SAML

Для того чтобы инициировать выполнение профиля, запрашивающая сторона SAML передает на оконечную точку службы запроса/изучения ответственного органа SAML сообщение `<AssertionIDRequest>`, `<SubjectQuery>`, `<AuthnQuery>`, `<AttributeQuery>` или `<AuthzDecisionQuery>`. Для определения местоположения этой оконечной точки и связей, поддерживаемых ответственным органом SAML, могут использоваться метаданные.

Запрашивающая сторона SAML должна использовать синхронную связь, такую как SOAP (см. раздел 10), для передачи запроса непосредственно провайдеру идентификации. Запрашивающая сторона должна аутентифицировать себя для ответственного органа SAML либо, подписав сообщение, либо используя любые другие механизмы, поддерживаемые связью.

Определяемые профилем правила для содержания различных сообщений перечислены в 11.4.7.4.1.

#### 11.4.7.3.2 Сообщение <Response>, переданное ответственным органом SAML

Ответственный орган SAML должен обработать сообщение запроса или запроса на изучение, как определено в разделе 8. После того как сообщение обработано, или в случае появления ошибки, ответственный орган SAML должен возвратить запрашивающей стороне SAML сообщение `<Response>`, содержащее соответствующий код состояния для завершения обмена по протоколу SAML. Если запрос успешно обнаружил одно или несколько совпадающих подтверждений, то они будут также включены в ответ.

Отвечающая сторона должна аутентифицировать себя для запрашивающей стороны или, подписав `<Response>`, либо используя любые другие механизмы, поддерживаемые связью.

Определяемые профилем правила для содержания сообщения `<Response>` перечислены в 11.4.7.4.2.

### 11.4.7.4 Использование протокола запроса/изучения

В настоящем разделе определяются оконечная точка протокола запроса/изучения на стороне ответственного органа SAML, куда доставляются сообщения запросов.

#### 11.4.7.4.1 Использование запроса/изучения

Элемент `<Issuer>` должен быть представлен.

Запрашивающая сторона должна аутентифицировать себя для отвечающей стороны и гарантировать целостность сообщения, подписав сообщение, либо используя любые другие механизмы, поддерживаемые связью.

#### 11.4.7.4.2 Использование сообщения <Response>

Элемент `<Issuer>` должен быть представлен и должен содержать уникальный идентификатор отвечающего ответственного органа SAML; атрибут `Format` должен быть пропущен или должен иметь значение `urn:oasis:names:tc:SAML:2.0:nameid-format:entity`. Это значение не обязательно должно совпадать с элементом `<Issuer>` в возвращаемом(ых) подтверждении(ях).

Отвечающая сторона должна аутентифицировать себя для запрашивающей стороны и гарантировать целостность сообщения, подписав сообщение, либо используя любые другие механизмы, поддерживаемые связью.

### 11.4.7.5 Использование метаданных

В разделе 9 определяется несколько элементов оконечной точки `<md:AssertionIDRequestService>`, `<md:AuthnQueryService>`, `<md:AttributeService>` и `<md:AuthzService>`, требуемых для описания поддерживаемых связей и мест(а) размещения, куда запрашивающая сторона может передавать запросы и запросы на изучение, используя этот профиль.

Ответственный орган SAML, если он шифрует результирующие подтверждения или содержание подтверждений для конкретного элемента, может использовать блок `<md:KeyDescriptor>` этого элемента с атрибутом `encryption` для определения соответствующего алгоритма шифрования и применяемых установок вместе с открытым ключом при доставке ключей шифрования.

Дескрипторы различных ролей могут содержать элементы `<md:NameIDFormat>`, `<md:AttributeProfile>` и `<saml:Attribute>` (при необходимости) для указания общей возможности поддерживать конкретные форматы идентификаторов имени, профили атрибутов, или конкретные атрибуты и значения. Способность поддерживать любые такие возможности во время данного запроса зависит от политики и остается на усмотрение ответственного органа.

## 11.4.8 Профиль преобразования идентификатора имени

В разделе 8.2.6 определяется протокол преобразования идентификатора имени для сопоставления идентификатора имени клиента с различными идентификаторами имени для того же клиента. Этот профиль описывает использование этого протокола с синхронной связью, такой как SOAP, определенной в разделе 10, и дополнительные рекомендации для защиты секретности клиента при помощи шифрования и ограничения использования преобразованного идентификатора.

### 11.4.8.1 Требуемая информация

**Идентификация**: `urn:oasis:names:tc:SAML:2.0:profiles:nameidmapping`

**Контактная информация**: security-services-comment@lists.oasis-open.org

**Описание**: Приводится ниже.

**Обновления**: Нет.

### 11.4.8.2 Обзор профиля

Правила обмена сообщениями и базовые правила обработки, которые управляют работой этого профиля, довольно широко представлены в разделе 8, где определены сообщения, которые должны передаваться в комбинации со связью, используемой для передачи сообщений. В разделе 10 определяется связь для обмена сообщениями как SOAP V1.1. Если в настоящей Рекомендации специально не указано иного, действуют все требования, приведенные в данной спецификации.

Рисунок 11-7 показан базовый формат для профиля преобразования идентификатора имени.

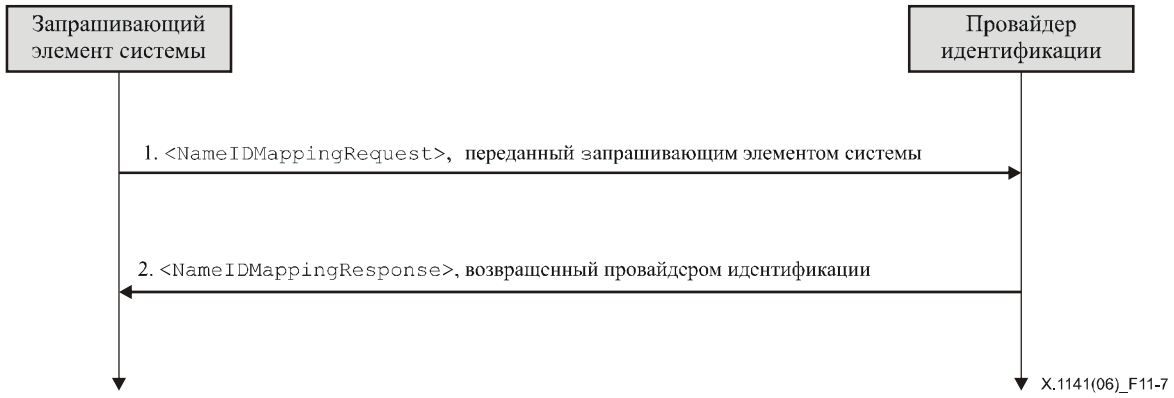


**Рисунок 11-7/X.1141 – Базовый формат для профиля преобразования идентификатора имени**

Профилем описываются следующие этапы:

1) **<NameIDMappingRequest> , переданный запрашивающим элементом**

   На этапе 1 запрашивающая сторона инициирует выполнение профиля, передавая <NameIDMappingRequest>.

2) **<NameIDMappingResponse>, переданный провайдером идентификации**

   На этапе 2 отвечающий провайдер идентификации (после обработки запроса) передает запрашивающей стороне сообщение <NameIDMappingResponse>.

#### 11.4.8.3 Описание профиля

В настоящем разделе используется услуга преобразования идентификатора имени, которая является на стороне провайдера идентификации, на которую доставляются сообщения <NameIDMappingRequest>.

##### 11.4.8.3.1 NameIDMappingRequest>, переданный запрашивающим элементом

Для того чтобы инициировать выполнение профиля, запрашивающая сторона передает сообщение <NameIDMappingRequest> на оконечную точку протокола преобразования идентификатора имени провайдера идентификации. Для определения местоположения этой оконечной точки и связей, поддерживаемых провайдером идентификации, могут использоваться метаданные.

Запрашивающая сторона должна использовать синхронную связь, такую как SOAP (см. раздел 10), для передачи запроса непосредственно провайдеру идентификации. Запрашивающая сторона должен аутентифицировать себя провайдеру идентификации, подписав <NameIDMappingRequest> либо используя любые другие механизмы, поддерживаемые связью.

Определяемые профилем правила для содержания сообщения <NameIDMappingRequest> перечислены в 11.4.8.4.1.

##### 11.4.8.3.2 <NameIDMappingResponse>, переданный провайдером идентификации

Провайдер идентификации должен обработать сообщение <ManageNameIDRequest>, как определено в разделе 8. После того как сообщение обработано, или в случае появления ошибки, провайдер идентификации должен возвратить запрашивающей стороне сообщение <NameIDMappingResponse>, содержащее соответствующий код состояния для завершения обмена по протоколу SAML.

Отвечающая сторона должна аутентифицировать себя для запрашивающей стороны, либо подписав <NameIDMappingResponse>, либо используя любые другие механизмы, поддерживаемые связью.

Определяемые профилем правила для содержания сообщения <NameIDMappingResponse> перечислены в 11.4.8.4.2.

#### 11.4.8.4 Использование протокола преобразования идентификатора имени

В разделе 8 определяется протокол преобразования идентификатора имени для преобразования идентификатора имени клиента в различные идентификаторы имени для того же самого клиента. В настоящем разделе описывается использование этого протокола и дополнительные рекомендации по защите секретности клиента, такие как ограничение использования преобразованного идентификатора.

##### 11.4.8.4.1 Использование запроса <NameIDMappingRequest>

Элемент <Issuer> должен быть представлен.

Запрашивающая сторона должна аутентифицировать себя для отвечающей стороны и гарантировать целостность сообщения, подписав сообщение, либо используя любые другие механизмы, поддерживаемые связью.

#### 11.4.8.4.2 Использование ответа <NameIDMappingResponse>

Элемент `<Issuer>` должен быть представлен и должен содержать уникальный идентификатор отвечающего провайдера идентификации; атрибут `Format` должен быть пропущен или должен иметь значение `urn:oasis:names:tc:SAML:2.0:nameid-format:entity`.

Отвечающая сторона должна аутентифицировать себя для запрашивающей стороны и гарантировать целостность сообщения, подписав сообщение, либо используя любые другие механизмы, поддерживаемые связью.

Правила шифрования W3C (см. 2.2.3) определяют использование шифрования для обеспечения конфиденциальности идентификатора имени. В большинстве случаев, провайдер идентификации должен зашифровать преобразованный идентификатор имени, который он возвращает запрашивающей стороне для защиты секретности клиента. Запрашивающая сторона может выделить элемент `<EncryptedID>` и поместить его в следующие протокольные сообщения или подтверждения.

**Ограничения на использование преобразованного идентификатора**

Дополнительные ограничения на использование результирующего идентификатора могут быть наложены провайдером идентификации при помощи возвращения преобразованного идентификатора имени в форме подтверждения `<Assertion>`, содержащего этот идентификатор в своем элементе `<Subject>`, но без каких-либо утверждений. Подтверждение затем шифруется, и результат используется как элемент `<EncryptedData>` в `<EncryptedID>`, возвращаемом запрашивающей стороне. Подтверждение может включить элемент `<Conditions>` для ограничения использования, как определено в разделе 8, например, временные ограничения или указание использовать определенные доверяющие стороны, и для защиты целостности оно должно быть подписано.

### 11.4.8.5 Использование метаданных

В настоящем разделе определяется элемент оконечной точки `<md:NameIDMappingService>`, для описания поддерживаемых связей и мест положения, требуемых для описания поддерживаемых связей и мест(а) размещения, куда запрашивающая сторона может передавать запросы, используя этот профиль.

Провайдер идентификации, если он шифрует результирующий идентификатор для определенного элемента, может использовать этот элемент блока `<md:KeyDescriptor>`, в котором атрибут `use` имеет значение `encryption`, для определения соответствующего алгоритма шифрования и установок, которые должны использоваться, вместе с открытым ключом, который должен использоваться при доставке ключей шифрования.

## 11.4.9 Профили атрибутов SAML

Профили атрибутов содержат определения, необходимые для ограничения возможностей выражения атрибутов SAML при работе с определенными типами информации атрибутов или при взаимодействии с внешними системами, требующими более строгих ограничений. В настоящем подразделе определяется базовый профиль атрибута SAML X.500/LDAP, профили UUID и профиль XACML.

### 11.4.9.1 Базовый профиль атрибута

Профиль атрибута `Basic` определяет упрощенную, но не уникальную систему обозначений атрибутов SAML вмесите со значениями атрибутов на основе встроенных типов данных, соответствующих документу W3C Datatypes, что исключает необходимость иметь схемы расширения для подтверждения синтаксиса.

**Требуемая информация**

**Идентификация**: `urn:oasis:names:tc:SAML:2.0:profiles:attribute:basic`

**Контактная информация**: security-services-comment@lists.oasis-open.org

**Описание**: Приводится ниже.

**Обновления**: Нет.

**Система обозначений атрибутов SAML**

Атрибут XML `NameFormat` в элементах `<Attribute>` должен иметь значение: `urn:oasis:names:tc:SAML:2.0:attrname-format:basic`.

Атрибут XML `Name` должен соответствовать правилам для этого формата, как определено в разделе 8.

– **Сравнение имени атрибута**

  Два элемента `<Attribute>` указывают один и тот же атрибут SAML, если и только если значения их атрибутов XML `Name` равны (в том смысле, как это описано в разделе 8).

**Определяемые профилем атрибуты XML**

Никаких дополнительных атрибутов XML не определяется для использования с элементом `<Attribute>`.

**Значения атрибутов SAML**

Тип схемы содержания элемента `<AttributeValue>` должен быть выбран из типов, определенных в Приложении А. Атрибут `xsi:type` должен быть представлен и должен иметь соответствующее значение.

**Пример**

```
<saml:Attribute NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-
format:basic"
    Name="FirstName">
 <saml:AttributeValue xsi:type="xs:string">By-Tor</saml:AttributeValue>
</saml:Attribute>
```

### 11.4.9.2 Профиль атрибута X.500/LDAP

Директории, основанные на Рекомендациях МСЭ-Т серии X.500 и документе IETF RFC 3377 широко распространены. Схема директории используется для моделирования информации, которая должна сохраняться в этих директориях. В частности, в атрибуте X.500 тип "Определения" используется для определения синтаксиса и других свойств атрибутов, базовой единицы хранения информации в директории (в настоящей Рекомендации это называется "атрибуты директории"). Типы атрибутов директории определяются в схеме самих спецификаций X.500 и LDAP, схеме других общедоступных документов (таких как схема inetOrgperson (см. IETF RFC 2798)), и схемах, определенных для частных целей. В любом из этих случаев, тем, кто реализует схему, полезно использовать преимущества этих типов атрибутов директории в контексте утверждений атрибутов SAML, обычно не создавая для них вручную определяемого в SAML атрибута Определения, и выполняя это в режиме взаимодействия.

Профиль атрибута X.500/LDAP определяет общую систему обозначений и представления этих атрибутов, выражая их как атрибуты SAML.

**Требуемая информация**

**Идентификация**: `urn:oasis:names:tc:SAML:2.0:profiles:attribute:X500` (это также является целевой областью имен, назначенной в соответствующей схеме профиля X.500/LDAP в Приложении А).

**Контактная информация**: security-services-comment@lists.oasis-open.org

**Описание**: Приводится ниже.

**Обновления**: Нет.

**Система обозначений атрибутов SAML**

Атрибут XML `NameFormat` в элементах `<Attribute>` должен иметь значение `urn:oasis:names:tc:SAML:2.0:attrname-format:uri`.

Для создания имен атрибутов используется область имен URN `oid`, описанная в IETF RFC 3061. При таком подходе атрибут XML `Name` основан на идентификаторе объекта, назначенном типу атрибута директории.

Пример:

`urn:oid:2.5.4.3`

Поскольку в процедурах X.500 требуется, чтобы каждый тип атрибута был бы идентифицирован уникальным идентификатором объекта, эта схема обозначений гарантирует, что полученные имена атрибутов SAML являются однозначными.

Для обеспечения возможности чтения человеком, в некоторых приложениях может существовать требование передавать вместе с OID URN дополнительное название строки (как определено в IETF RFC 3061). Для этой цели может использоваться дополнительный атрибут XML `FriendlyName` (определенный в разделе 8). Если определение типа атрибута директории включает в себя один или несколько дескрипторов (коротких имен) для данного типа атрибута, значением `FriendlyName`, если оно представлено, должен быть один из определенных дескрипторов.

Два элемента `<Attribute>` обозначают один и тот же атрибут SAML, если и только если значения их атрибутов XML `Name` равны в том смысле, как это описано в IETF RFC 3061. Атрибут `FriendlyName` не играет при сравнении никакой роли.

**Определяемые профилем атрибуты XML**

Никаких дополнительных атрибутов XML не определяется для использования с элементом `<Attribute>`.

**Значения атрибутов SAML**

Атрибут директории типа Определения для использования в директориях протокола X.500 определяют синтаксис атрибута с использованием ASN.1. Для использования в протоколе LDAP, атрибут директории Определения дополнительно содержит синтаксис протокола LDAP, который определяет каким образом значения атрибута или подтверждения, соответствующие этому синтаксису, должны быть представлены в процессе

передачи в протоколе LDAP (известном как кодирование, используемое в протоколе LDAP). В процессе кодирования, используемого в протоколе LDAP, обычно выполняется кодировка Unicode символов в форме UTF-8. Этот профиль атрибута SAML определяет форму значений атрибутов SAML только для тех атрибутов директории, которые имеют синтаксис LDAP. Будущие расширения этого профиля могут определять значения форматов атрибутов для директории, синтаксис которых определяет другие варианты кодирования.

Для того чтобы представить правила кодирования, используемые для конкретного значения атрибута, элемент `<AttributeValue>` должен содержать атрибут XML `Encoding`, определенный в области имен XML `urn:oasis:names:tc:SAML:2.0:profiles:attribute:X500`.

Для любого атрибута директории с синтаксисом, который определяет кодирование, используемое в протоколе LDAP, и формирует в качестве значений только строки символов UTF-8, само значение атрибута SAML кодируется в виде простой строки UTF-8, содержащей элемент `<AttributeValue>` без дополнительных пробелов. В таких ситуациях атрибут XML `xsi:type` должен быть установлен в значение **xs:string**. Определяемый профилем атрибут XML `Encoding` содержит значение `LDAP`.

Список некоторых атрибутов синтаксиса LDAP (и связанных с ними идентификаторов OID), к которым применяется это правило, имеет вид:

| | |
|---|---|
| Описание типа атрибута | 1.3.6.1.4.1.1466.115.121.1.3 |
| Строка битов | 1.3.6.1.4.1.1466.115.121.1.6 |
| Булево значение | 1.3.6.1.4.1.1466.115.121.1.7 |
| Строка с обозначением страны | 1.3.6.1.4.1.1466.115.121.1.11 |
| DN | 1.3.6.1.4.1.1466.115.121.1.12 |
| Строка с обозначением директории | 1.3.6.1.4.1.1466.115.121.1.15 |
| Номер телефона/факса | 1.3.6.1.4.1.1466.115.121.1.22 |
| Обобщенное время | 1.3.6.1.4.1.1466.115.121.1.24 |
| Строка IA5 | 1.3.6.1.4.1.1466.115.121.1.26 |
| ЦЕЛОЕ ЧИСЛО | 1.3.6.1.4.1.1466.115.121.1.27 |
| Описание синтаксиса LDAP | 1.3.6.1.4.1.1466.115.121.1.54 |
| Описание правил согласования | 1.3.6.1.4.1.1466.115.121.1.30 |
| Описание применения правил согласования | 1.3.6.1.4.1.1466.115.121.1.31 |
| Имя и дополнительный идентификатор UID | 1.3.6.1.4.1.1466.115.121.1.34 |
| Описание формы имени | 1.3.6.1.4.1.1466.115.121.1.35 |
| Цифровая строка | 1.3.6.1.4.1.1466.115.121.1.36 |
| Описание класса объекта | 1.3.6.1.4.1.1466.115.121.1.37 |
| Строка байтов | 1.3.6.1.4.1.1466.115.121.1.40 |
| OID | 1.3.6.1.4.1.1466.115.121.1.38 |
| Другой почтовый ящик | 1.3.6.1.4.1.1466.115.121.1.39 |
| Почтовый адрес | 1.3.6.1.4.1.1466.115.121.1.41 |
| Адрес местонахождения | 1.3.6.1.4.1.1466.115.121.1.43 |
| Печатаемая строка | 1.3.6.1.4.1.1466.115.121.1.44 |
| Подтверждение элемента цепочки | 1.3.6.1.4.1.1466.115.121.1.58 |
| Номер телефона | 1.3.6.1.4.1.1466.115.121.1.50 |
| Универсальное скоординированное время (UTC) | 1.3.6.1.4.1.1466.115.121.1.53 |

Для все других синтаксисов LDAP значение атрибута кодируется как содержание элемента `<AttributeValue>` в кодировке base64, охватывая значение атрибута ASN.1, кодированного согласно протоколу LDAP как строка байтов. Атрибут XML `xsi:type` должен быть установлен в **xs:base64Binary**. Определяемый профилем атрибут XML `Encoding` содержит значение "`LDAP`".

Во время сравнения значений атрибутов SAML с целью проверки их равенства, должны применяться правила согласования, определенные для соответствующего типа атрибута директории (например, чувствительность к состоянию регистра).

**Схема, определяемая профилем**

Приведенный далее листинг схемы показывает определяемый профилем атрибут XML `Encoding` (см. Приложение A):

```
<schema
    targetNamespace="urn:oasis:names:tc:SAML:2.0:profiles:attribute:X500"
    xmlns="http://www.w3.org/2001/XMLSchema"
    elementFormDefault="unqualified"
    attributeFormDefault="unqualified"
    blockDefault="substitution"
    version="2.0">
    <annotation>
        <documentation>
            Document identifier: saml-schema-x500-2.0
            Location: http://docs.oasis-open.org/security/saml/v2.0/
```

```
            Revision history:
              V2.0 (March, 2005):
                Custom schema for X.500 attribute profile, first published
in SAML 2.0.
        </documentation>
    </annotation>
    <attribute name="Encoding" type="string"/>
</schema>
```

**Пример**

Далее приведен пример преобразования атрибута директории "givenName", представляющего собой подтверждение имени объекта SAML. Его идентификатор объекта имеет вод {joint-iso-itu-t(2) ds(5) attributeType(4) givenName(42)} и его синтаксис LDAP – это Строка директории (Directory String):

```
<saml:Attribute
xmlns:x500="urn:oasis:names:tc:SAML:2.0:profiles:attribute:X500"
    NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:uri"
    Name="urn:oid:2.5.4.42" FriendlyName="givenName">
 <saml:AttributeValue xsi:type="xs:string"
    x500:Encoding="LDAP">Steven</saml:AttributeValue>
</saml:Attribute>
```

### 11.4.9.3 Профиль атрибута UUID

Профиль атрибута UUID (универсальный уникальный идентификатор) стандартизует выражение значений UUID в виде имен и значении атрибутов SAML. Он применяется, когда система источников атрибута является системой, которая идентифицирует атрибут или его значение при помощи UUID.

**Требуемая информация**

**Идентификация**: `urn:oasis:names:tc:SAML:2.0:profiles:attribute:UUID`

**Контактная информация**: security-services-comment@lists.oasis-open.org

**Описание**: Приводится ниже.

**Обновления**: Нет.

**Система обозначений атрибутов SAML**

Атрибут XML `NameFormat` в элементах `<Attribute>` должен иметь значение `urn:oasis:names:tc:SAML:2.0:attrname-format:uri`.

Если базовым представлением имени атрибута является UUID, тогда используется область имен URN `uuid`, описанная в Рекомендации МСЭ-Т X.667. При таком подходе атрибут XML `Name` основан на форме URN базового представления UUID, который идентифицирует атрибут.

Пример:

```
urn:uuid:f81d4fae-7dec-11d0-a765-00a0c91e6bf6
```

Если базовым представлением имени атрибута является не UUID, тогда в атрибуте XML `Name` может использоваться любая форма URI.

Для обеспечения удобочитаемости для человека, для некоторых приложений может быть установлено требование передавать вместе с URI дополнительную строку. Для этой цели может использоваться дополнительный атрибут XML `FriendlyName`.

Два элемента `<Attribute>` обозначают один и тот же атрибут SAML, если и только если значения их атрибутов XML `Name` равны в том смысле, как это описано в Рекомендации МСЭ-Т X.667. Атрибут `FriendlyName` не играет при сравнении никакой роли.

**Определяемые профилем атрибуты XML**

Никаких дополнительных атрибутов XML не определяется для использования с элементом `<Attribute>`.

**Значения атрибутов SAML**

В тех случаях, когда значением атрибута также является UUID, тот же самый синтаксис, что использован выше, должен использоваться для выражения значения элемента `<AttributeValue>`. Атрибут XML `xsi:type` должен быть установлен в значение **xs:anyURI**.

Если значением атрибута не является значение UUID, тогда не никаких ограничений на использование элемента `<AttributeValue>`.

**Пример**

Далее приведен пример атрибута DCE расширенного регистра, а именно, установки "pre_auth_req", которая имеет хорошо известный универсальный уникальный идентификатор UUID = 6c9d0ec8-dd2d-11cc-abdd-080009353559 и имеет целочисленное значение.

```
<saml:Attribute NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:uri"
    Name="urn:uuid:6c9d0ec8-dd2d-11cc-abdd-080009353559"
    FriendlyName="pre_auth_req">
 <saml:AttributeValue xsi:type="xs:integer">1</saml:AttributeValue>
</saml:Attribute>
```

#### 11.4.9.4 Профиль атрибута XACML

Подтверждения атрибута SAML могут использоваться как входные сигналы для решений об авторизации, принимаемых в соответствии с Рек. МСЭ-Т X.1142. Поскольку формат атрибута SAML отличается от формата атрибута XACML, должно быть выполнено преобразование. Профиль атрибута XACML упрощает это преобразование при помощи стандартизации обозначений, значения синтаксиса и дополнительных метаданных атрибута. Атрибуты SAML, созданные в соответствие с этим профилем, могут быть автоматически преобразованы в атрибуты XACML и использованы как входные сигналы для решений об авторизации XACML.

**Требуемая информация**

**Идентификация**: `urn:oasis:names:tc:SAML:2.0:profiles:attribute:XACML` (это также представляет собой целевую область имен, назначенную в соответствующей схеме профиля XACML в Приложении A).

**Контактная информация**: security-services-comment@lists.oasis-open.org

**Описание**: Приводится ниже.

**Обновления**: Нет.

**Система обозначений атрибутов SAML**

Атрибут XML `NameFormat` в элементах `<Attribute>` должен иметь значение `urn:oasis:names:tc:SAML:2.0:attrname-format:uri`.

Атрибут XML `Name` должен соответствовать правилам, определенным для этого формата, как определено в разделе 8.

Для обеспечения удобочитаемости для человека, для некоторых приложений может быть установлено требование передавать URN вместе с OID. Для этой цели может использоваться дополнительный атрибут XML `FriendlyName` (определенный в разделе 8), но они не может быть переведен в эквивалент атрибута XACML.

Два элемента `<Attribute>` обозначают один и тот же атрибут SAML, если и только если значения их атрибутов XML `Name` равны при побитовом сравнении. Атрибут `FriendlyName` не играет при сравнении никакой роли.

**Определяемые профилем атрибуты XML**

Расширяемый язык разметки, предусматривающий контроль доступа (XACML), требует, чтобы каждый атрибут содержал явный тип данных. Для получения этого значения типа данных в области имен XML `urn:oasis:names:tc:SAML:2.0:profiles:attribute:XACML` определен новый элемент со значением URI, имеющий название `DataType`.

Элементы SAML `<Attribute>`, соответствующие этому профилю, должны содержать определяемый в области имен атрибут `DataType` или их значение считается равным `http://www.w3.org/2001/XMLSchema#string`.

Если используются нестандартные значения, тогда каждая точка принятия стратегического решения PDP языка XACML, в которой будут использоваться преобразованные атрибуты SAML с нестандартными значениями `DataType`, должны быть расширены для поддержки этих новых типов данных.

**Значения атрибутов SAML**

Синтаксис содержания элемента `<AttributeValue>` должен соответствовать типу данных, указанных в определяемом профилем атрибуте XML `DataType`, находящемся в родительском элементе `<Attribute>`. Для типов данных, соответствующих типам, определенным в разделе 8, в элементе(ах) `<AttributeValue>` должен также использоваться атрибут XML `xsi:type`.

**Схема, определяемая профилем**

Приведенный далее листинг схемы показывает, как определяется определяемый профилем атрибут `DataType XML` (Приложение A):

```
<schema
    targetNamespace="urn:oasis:names:tc:SAML:2.0:profiles:attribute:XACML"
    xmlns="http://www.w3.org/2001/XMLSchema"
    elementFormDefault="unqualified"
    attributeFormDefault="unqualified"
    blockDefault="substitution"
    version="2.0">
    <annotation>
        <documentation>
            Document identifier: saml-schema-xacml-2.0
            Location: http://docs.oasis-open.org/security/saml/v2.0/
            Revision history:
            V2.0 (March, 2005):
              Custom schema for XACML attribute profile, first published in
SAML 2.0.
        </documentation>
    </annotation>
    <attribute name="DataType" type="anyURI"/>
</schema>
```

**Пример**

Далее приведен пример преобразования атрибута протокола LDAP/X.500 "givenName", представляющего собой подтверждение имени объекта SAML. На примере также показано, что один атрибут SAML может соответствовать многим профилям атрибутов, если они совместимы друг с другом.

```
<saml:Attribute
xmlns:xacmlprof="urn:oasis:names:tc:SAML:2.0:profiles:attribute:XACML"
 xmlns:ldapprof="urn:oasis:names:tc:SAML:2.0:profiles:attribute:LDAP"
    xacmlprof:DataType="http://www.w3.org/2001/XMLSchema#string"
    ldapprof:Encoding="LDAP"
    NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:uri"
    Name="urn:oid:2.5.4.42" FriendlyName="givenName">
 <saml:AttributeValue xsi:type="xs:string">By-Tor</saml:AttributeValue>
</saml:Attribute>
```

ПРИМЕЧАНИЕ (информативное). – PE39 (см. OASIS PE:2006) разъясняет вышеприведенный пример следующим образом:

```
<saml:Attribute
  xmlns:xacmlprof="urn:oasis:names:tc:SAML:2.0:profiles:attribute:XACML"
      xmlns:ldapprof="urn:oasis:names:tc:SAML:2.0:profiles:AttributeValue:LDAP"
           xacmlprof:DataType="http://www.w3.org/2001/XMLSchema#string"
           ldapprof:Encoding="LDAP"
           NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:uri"
           Name="urn:oid:2.5.4.42" FriendlyName="givenName">
      <saml:AttributeValue xsi:type="xs:string">By-Tor</saml:AttributeValue>
</saml:Attribute>
```

## 12 Правила аутентификации SAML

Настоящая Рекомендация определяет синтаксис для определения объявлений правил аутентификации и первоначальный перечень классов контекстов аутентификации.

### 12.1 Концепция правил аутентификации

Если доверяющая сторона должна доверять аутентификации клиента, выполненной органом по аутентификации, то доверяющая сторона может потребовать дополнительно к самому подтверждению некоторую информацию для того, чтобы оценить уровень доверия, с которым она может воспринимать это подтверждение. В настоящей Рекомендации определяется схема XML для создания Объявлений правил аутентификации – документов XML, которые позволяют органу по аутентификации представить доверяющей стороне эту дополнительную информацию. Кроме того, в настоящей Рекомендации определяется множество классов контекстов аутентификации; категорий, на которое подразделятся различные Объявление правил аутентификации, что упрощает их понимание.

SAML не предписывает какой-либо единственной технологии, протокола, или правил для процедур, при помощи которых органы по аутентификации формируют идентификаторы для клиентов и при помощи которых эти клиенты затем аутентифицируют себя для органов по аутентификации. Различные органы по аутентификации будут использовать различные технологии, выполнять различные процедуры, и будут связаны различными правовыми обязательствами в отношении того, как они аутентифицируют клиентов.

Выбор, который делает орган по аутентификации, будет во многом обусловлен требованиями доверяющих сторон, с которыми взаимодействует орган по аутентификации. Сами эти требования будут обусловлены видом услуги (т. е. чувствительностью передаваемой информации, связанными сними финансовыми параметрами, границами риска доверяющих сторон и т. д.), которую доверяющая сторона будет предоставлять клиенту.

Следовательно, для всего остального, кроме обычных услуг, если доверяющая сторона должна иметь высокую степень уверенности в достоверности подтверждений аутентификации, которые она принимает от органа по аутентификации, для нее будет необходимо знать, какие технологии, протоколы и процедуры были использованы или реализованы в исходном механизме аутентификации, на котором основано подтверждение аутентификации. Имея эту информацию и доверяя источнику исходного подтверждения, доверяющая сторона будет лучше способна принимать обоснованные решения о наименованиях относительно того, каким к каким услугам может быть разрешен доступ объектам подтверждения аутентификации.

Контекст аутентификации определяется как информация, дополнительная к самому подтверждению аутентификации, который доверяющая сторона может потребовать прежде, чем принять решение о наименовании в отношении какого-либо подтверждения аутентификации. Такой контекст может включать в себя, но не ограничиваться реальным используемым методом аутентификации.

### 12.2 Объявление правил аутентификации

Если доверяющая сторона должна доверить аутентификация другого элемента, подтверждаемой органом по аутентификации, то доверяющая сторона может потребовать дополнительно к самому подтверждению некоторую информацию, которая даст ей возможность ввести аутентификацию в контекст управления рисками. Такая информация может включать в себя:

- механизмы идентификации инициирующего пользователя (например, один на один, он-лайн, общие секреты);
- механизмы, минимизирующие возможность дискредитации сведений, подтверждающих подлинность (например, частота возобновления сведений, подтверждающих подлинность, генерация ключа на стороне клиента);
- механизмы для хранения и защиты сведений, подтверждающих подлинность (например, смарт-карты, пароли);
- механизм или метод аутентификации (например, пароль).

Варианты и перестановки перечисленных выше характеристик гарантирует, что не все подтверждения аутентификации будут одинаковы в отношении того, насколько будет им доверять доверяющая сторона; определенное подтверждение аутентификации будет характеризоваться значениями каждой из этих (и других) переменных.

Орган по аутентификации SAML может доставить доверяющей стороне дополнительную контекстную информации аутентификации в форме объявления правил аутентификации, т. е. документа XML, который либо введен непосредственно в него, либо указан в подтверждении аутентификации, которое орган по аутентификации представляет доверяющей стороне.

Запрашивающие стороны SAML имеют возможность потребовать, чтобы аутентификация соответствовала определенному контексту аутентификации, указав этот контекст в запросе на аутентификацию. Запрашивающая сторона может также определить, что аутентификация должна быть представлена с контекстом аутентификации, который *превосходит* некоторое объявленное значение (для некоторых согласованных определений понятия "превосходит").

#### 12.2.1 Модель данных

Конкретное объявление правил аутентификации, определенное в настоящей Рекомендации, буде получать характеристики процессов, процедур и механизмов, при помощи которых орган по аутентификации, проверяющий объект аутентификации, прежде чем сформировать его идентификатор, защищает секреты, на которых основана последующая аутентификация, и механизмы, используемые для аутентификации. Эти характеристики в контексте схемы аутентификации классифицируются следующим образом.

- Идентификация – характеристики, которые описывают процессы и механизм, используемые органом по аутентификации для первоначального создания связи между объектом и его идентификатором (или именем), при помощи которого будет опознаваться этот объект.
- Техническая защита – характеристики, которые описывают, как обеспечивается безопасность "секрета" (знаний или сведений, которые позволяют аутентифицировать объект для органа по аутентификации).
- Эксплуатационная защита – характеристики, которые описывают процедуры обеспечения безопасности, используемые органом по аутентификации (например, проверка безопасности, архивирование записей).

- Метод аутентификации – характеристики, которые определяют механизмы, при помощи которых объект переданного подтверждения аутентифицирует себя для органа по аутентификации (например, пароль и смарт-карта).
- Руководящие соглашения – характеристики, которые описывают юридические основы (например, законность ограничений и договорных обязательств), определяющих событие аутентификации и/или связанную с ним инфраструктуру технической аутентификации.

### 12.2.2 Расширяемость

Схема объявления правил аутентификации обладает точками расширения, точно определенными в элементе `<Extension>`. Органы по аутентификации могут использовать этот элемент для введения дополнительных подробностей контекста аутентификации в подтверждения SAML, которые они формируют (предполагается, что использующая их доверяющая сторона способна понять эти расширения). Эти дополнительные элементы должны быть в отдельной области имен XML, на которых основаны объявления правил аутентификации или схема классов, которые используются в самом объявлении.

### 12.2.3 Правила обработки

Дополнительные правила обработки объявлений правил аутентификации определены в разделе 8, эти правила обработки представляют собой варианты реализации, использующие общие интерпретации относительной силы и качества конкретных объявлений правил аутентификации и не могут быть выражены в абсолютных значениях или представлены виде правил, которые должны соблюдаться в этих вариантах реализации.

### 12.2.4 Схема

Настоящий раздел не является нормативным.

Листинг полного контекста аутентификации типов схемы XML и контекста аутентификации самой схемы XML, используемый для подтверждения достоверности отдельных общих объявлений, приведена в Дополнении VI.

## 12.3 Классы контекстов аутентификации

Количество перестановок различных характеристик гарантирует, что теоретически существует бесконечное число уникальных вариантов контекстов аутентификации. То есть в теории, ожидается, что любая конкретная доверяющая сторона будет способна выполнить синтаксический разбор произвольного объявления правил аутентификации и, что важнее, проанализировать заявление для того, чтобы оценить "качество" соответствующих подтверждений аутентификации. Выполнение таких оценок – задача не тривиальная.

К счастью, возможна оптимизация. На практике многие контексты аутентификации попадают в категории, определенные промышленной практикой и технологией. Например, многие контексты аутентификации B2C веб-браузера будут (частично) определены клиентом, аутентифицирующим себя для органа по аутентификации путем представления пароля в ходе сеанса связи с защитой TLS. В корпоративном мире общим решением будет аутентификация на базе сертификатов. Несомненно, полный контекст аутентификации не ограничивается до деталей аутентификации отдельного клиента. Тем не менее, метод аутентификации часто является наиболее *заметной* характеристикой, и, раз это так, он может служить удобным классификатором для класса соответствующих контекстов аутентификации.

Эта концепция изложена в настоящей Рекомендации как определение набора *классов контекстов аутентификации*. Каждый класс определяет некоторое подмножество полного множества контекстов аутентификации. Классы выбраны как представители существующих методов и технологий аутентификации, и предоставляют подтверждающим и доверяющим сторонам удобное краткое условное обозначение для обозначения вопросов контекста аутентификации.

Например, орган по аутентификации может включить в себя полное объявление правил аутентификации, в соответствии с которыми он представляет доверяющей стороне подтверждение того, что данный контекст аутентификации также относится к классу контекстов аутентификации. Для некоторых доверяющих сторон это подтверждение является достаточным для того, чтобы они могли назначить определенному подтверждению аутентификации соответствующий уровень достоверности. Другие доверяющие стороны могут предпочесть выполнить полный анализ объявления правил аутентификации самостоятельно. Таким же образом, способность ссылаться на класс контекста аутентификации, а не перечислять все подробности данного объявления правил аутентификации, упростит способ, при помощи которого доверяющая сторона может сообщить органу по аутентификации о своих желаниях и/или потребностях.

### 12.3.1 Преимущества классов контекстов аутентификации

От введения дополнительного уровня классов и определение исходного списка представительных и гибких классов ожидается, что оно:

- упростит достижение соглашения между органом по аутентификации и доверяющей стороной о том, какие контексты аутентификации являются приемлемыми, поскольку даст им почву для обсуждения;
- упростит для доверяющих сторон указание предпочтений при запросе более высокого подтверждения аутентификации от органа по аутентификации;
- упростит для доверяющих сторон обработку объявления правил аутентификации, поскольку даст им возможность удовлетворить потребности при помощи соответствующего класса;

- оградит доверяющие стороны от воздействия новых технологий аутентификации;
- упростит для органов по аутентификации возможность объявлять о своих возможностях по аутентификации, например, при помощи WSDL.

### 12.3.2 Правила обработки

Другие правила обработки для классов контекстов аутентификации описываются в разделе 8. В большинстве случаев эти правила обработки относятся к вариантам реализации, использующим общую интерпретацию относительной силы или относительного качества конкретных классов контекстов аутентификации, и не могут быть выражены в абсолютных единицах или представлены в виде правил, которые должны выполняться в этих вариантах реализации.

### 12.3.3 Расширяемость

Как и базовая схема объявления правил аутентификации, схемы отдельных классов контекстов аутентификации допускают наличие элемента `<Extension>` в некоторых участках структуры дерева. Как правило, там, где элемент `<Extension>` появляется в виде дочернего элемента относительно элемента `<xs:choice>`, эта возможность устраняется при создании соответствующего определения схемы класса, являющейся ограничением базового типа. Когда элемент `<Extension>` появляется в виде дополнительного дочернего элемента относительно элемента `<xs:sequence>`, элемент `<Extension>` может сохраниться в дополнение к остальным необходимым элементам.

Следовательно, объявления правил аутентификации могут содержать элемент `<Extension>` (с дополнительными элементами в других областях имен) и все равно соответствовать схемам класса контекста аутентификации (если они отвечают остальным требованиям схемы, конечно же).

Схемы класса контекста аутентификации ограничивают тип Определения в базовой схеме контекста аутентификации. Являясь точкой расширения, схемы класса контекста аутентификации сами могут быть ограничены – их тип Определения, используется в качестве базовых типов в некоторых других схемах (возможно, определенных каким-либо сообществом, желающим более точно определить класс контекста аутентификации). Для предотвращения логического несоответствия, любая такая схема расширения может только еще больше огранить тип Определения для данной схемы класса. Для выполнения этого ограничения, схемы класса контекста аутентификации определяются с атрибутом `finalDefault="extension"` в элементе `<schema>` для предотвращения возможности получения этого типа.

### 12.3.4 Схемы

В последующих подразделах перечислены классы контекстов аутентификации. Классы перечислены в алфавитном порядке; никаких других правил не использовалось при расстановке классов. Те, кто реализует схемы, могут выбирать, какие классы поддерживать, как рекомендовано соответствующими руководствами в настоящей Рекомендации (см. раздел 13). Классы уникально идентифицируется посредством их URI со следующей основой:

```
urn:oasis:names:tc:SAML:2.0:ac:classes
```

Схемы класса определяются как ограничения, накладываемые на отдельные часть базовой схемы "типы" контекста аутентификации. Говорят, что элементы XML, которые подтверждаются на основании данной схемы класса контекста аутентификации, *соответствуют* данному классу контекста аутентификации.

Поскольку схема класса переопределяет элементы и типы и импортирует их в область имен схемы класса, то соответствующее классу объявление правил аутентификации не мгновенно подтверждается на основании базовой схемы контекста аутентификации.

#### 12.3.4.1 Протокол Интернет

**URI**: `urn:oasis:names:tc:SAML:2.0:ac:classes:InternetProtocol`

Этот URI используется также в качестве целевой области имен в соответствующей схеме класса контекста аутентификации в Приложении A.

Класс протокол Интернет применим, когда клиент аутентифицируется при помощи предоставленного ему IP-адреса.

```
<?xml version="1.0" encoding="UTF-8"?>

<xs:schema
  targetNamespace="urn:oasis:names:tc:SAML:2.0:ac:classes:InternetProtocol"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns="urn:oasis:names:tc:SAML:2.0:ac:classes:InternetProtocol"
  finalDefault="extension"
  blockDefault="substitution"
  version="2.0">

  <xs:redefine schemaLocation="saml-schema-authn-context-types-2.0.xsd">
```

```
    <xs:annotation>
      <xs:documentation>
        Class identifier: urn:oasis:names:tc:SAML:2.0:ac:classes:InternetProtocol
        Document identifier: saml-schema-authn-context-ip-2.0
        Location: http://docs.oasis-open.org/security/saml/v2.0/
        Revision history:
          V2.0 (March, 2005):
            New authentication context class schema for SAML V2.0.
      </xs:documentation>
    </xs:annotation>

    <xs:complexType name="AuthnContextDeclarationBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthnContextDeclarationBaseType">
          <xs:sequence>
            <xs:element ref="Identification" minOccurs="0"/>
            <xs:element ref="TechnicalProtection" minOccurs="0"/>
            <xs:element ref="OperationalProtection" minOccurs="0"/>
            <xs:element ref="AuthnMethod"/>
            <xs:element ref="GoverningAgreements" minOccurs="0"/>
            <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
          </xs:sequence>
          <xs:attribute name="ID" type="xs:ID" use="optional"/>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="AuthnMethodBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthnMethodBaseType">
          <xs:sequence>
            <xs:element ref="PrincipalAuthenticationMechanism" minOccurs="0"/>
            <xs:element ref="Authenticator"/>
            <xs:element ref="AuthenticatorTransportProtocol" minOccurs="0"/>
            <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="AuthenticatorBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthenticatorBaseType">
          <xs:sequence>
            <xs:element ref="IPAddress"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

  </xs:redefine>

</xs:schema>
```

#### 12.3.4.2 Класс InternetProtocolPassword

**URI**: `urn:oasis:names:tc:SAML:2.0:ac:classes:InternetProtocolPassword`

Отметим, что этот URI используется также в качестве целевой области имен в схеме соответствующего класса контекста аутентификации в Приложении A.

Класс InternetProtocolPassword применяется, когда клиент аутентифицируется при помощи использования предоставленного IP-адреса в дополнение к имени пользователя/паролю.

```
<?xml version="1.0" encoding="UTF-8"?>

<xs:schema
  targetNamespace="urn:oasis:names:tc:SAML:2.0:ac:classes:InternetProtocolPassword"
  xmlns:ac="urn:oasis:names:tc:SAML:2.0:ac"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns="urn:oasis:names:tc:SAML:2.0:ac:classes:InternetProtocolPassword"
  finalDefault="extension"
  blockDefault="substitution"
  version="2.0">
```

```
  <xs:redefine schemaLocation="saml-schema-authn-context-types-2.0.xsd">

    <xs:annotation>
      <xs:documentation>
        Class identifier:
urn:oasis:names:tc:SAML:2.0:ac:classes:InternetProtocolPassword
        Document identifier: saml-schema-authn-context-ippword-2.0
        Location: http://docs.oasis-open.org/security/saml/v2.0/
        Revision history:
          V2.0 (March, 2005):
            New authentication context class schema for SAML V2.0.
      </xs:documentation>
    </xs:annotation>

    <xs:complexType name="AuthnContextDeclarationBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthnContextDeclarationBaseType">
          <xs:sequence>
            <xs:element ref="Identification" minOccurs="0"/>
            <xs:element ref="TechnicalProtection" minOccurs="0"/>
            <xs:element ref="OperationalProtection" minOccurs="0"/>
            <xs:element ref="AuthnMethod"/>
            <xs:element ref="GoverningAgreements" minOccurs="0"/>
            <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
          </xs:sequence>
          <xs:attribute name="ID" type="xs:ID" use="optional"/>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="AuthnMethodBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthnMethodBaseType">
          <xs:sequence>
            <xs:element ref="PrincipalAuthenticationMechanism" minOccurs="0"/>
            <xs:element ref="Authenticator"/>
            <xs:element ref="AuthenticatorTransportProtocol" minOccurs="0"/>
            <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="AuthenticatorBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthenticatorBaseType">
          <xs:sequence>
            <xs:element ref="Password"/>
            <xs:element ref="IPAddress"/>
            <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

  </xs:redefine>

</xs:schema>
```

#### 12.3.4.3 Kerberos

**URI**: `urn:oasis:names:tc:SAML:2.0:ac:classes:Kerberos`

Этот URI используется также в качестве целевой области имен в схеме соответствующего класса контекста аутентификации в Приложении А.

Этот класс применяется, когда клиент аутентифицируется при помощи пароля для местного органа по аутентификации, с целью получения билета Kerberos. Этот билет Kerberos затем используется для последующей аутентификации в сети.

ПРИМЕЧАНИЕ 1. – Возможно, что в ходе аутентификации клиента орган по аутентификации укажет (при помощи данного класса контекста) тип данных предварительной аутентификации, который был использован центром распределения ключей Kerberos (IETF RFC 1510). Метод, который использует орган по аутентификации для получения этой информации, в настоящей Рекомендации не рассматривается, однако его использование настоятельно рекомендуется

для передачи органу по аутентификации данных предварительной аутентификации и других подробностей, связанных с системой Kerberos (например, данные о сроке действия билета).

```
<?xml version="1.0" encoding="UTF-8"?>

<xs:schema targetNamespace="urn:oasis:names:tc:SAML:2.0:ac:classes:Kerberos"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns="urn:oasis:names:tc:SAML:2.0:ac:classes:Kerberos"
  finalDefault="extension"
  blockDefault="substitution"
  version="2.0">

  <xs:redefine schemaLocation="saml-schema-authn-context-types-2.0.xsd">

    <xs:annotation>
      <xs:documentation>
        Class identifier: urn:oasis:names:tc:SAML:2.0:ac:classes:Kerberos
        Document identifier: saml-schema-authn-context-kerberos-2.0
        Location: http://docs.oasis-open.org/security/saml/v2.0/
        Revision history:
          V2.0 (March, 2005):
            New authentication context class schema for SAML V2.0.
      </xs:documentation>
    </xs:annotation>

    <xs:complexType name="AuthnContextDeclarationBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthnContextDeclarationBaseType">
          <xs:sequence>
            <xs:element ref="Identification" minOccurs="0"/>
            <xs:element ref="TechnicalProtection" minOccurs="0"/>
            <xs:element ref="OperationalProtection" minOccurs="0"/>
            <xs:element ref="AuthnMethod"/>
            <xs:element ref="GoverningAgreements" minOccurs="0"/>
            <xs:element ref="Extension" minOccurs="0"
maxOccurs="unbounded"/>
          </xs:sequence>
          <xs:attribute name="ID" type="xs:ID" use="optional"/>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="AuthnMethodBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthnMethodBaseType">
          <xs:sequence>
            <xs:element ref="PrincipalAuthenticationMechanism"/>
            <xs:element ref="Authenticator"/>
            <xs:element ref="AuthenticatorTransportProtocol" minOccurs="0"/>
            <xs:element ref="Extension" minOccurs="0"
maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="PrincipalAuthenticationMechanismType">
      <xs:complexContent>
        <xs:restriction base="PrincipalAuthenticationMechanismType">
          <xs:sequence>
              <xs:element ref="RestrictedPassword"/>
          </xs:sequence>
          <xs:attribute name="preauth" type="xs:integer" use="optional"/>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="AuthenticatorBaseType">
      <xs:complexContent>
```

```
        <xs:restriction base="AuthenticatorBaseType">
          <xs:sequence>
            <xs:element ref="SharedSecretChallengeResponse"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="SharedSecretChallengeResponseType">
      <xs:complexContent>
        <xs:restriction base="SharedSecretChallengeResponseType">
          <xs:attribute name="method" type="xs:anyURI"
fixed="urn:oasis:names:tc:SAML:2.0:ac:classes:Kerberos"/>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

  </xs:redefine>

</xs:schema>
```

Далее приведен пример элемента XML, соответствующего схеме этого класса:

```
<AuthenticationContextDeclaration
  xmlns="urn:oasis:names:tc:SAML:2.0:ac:classes:Kerberos">

  <AuthnMethod>

    <PrincipalAuthenticationMechanism preauth="0">
      <RestrictedPassword>
        <Length min="4"/>
      </RestrictedPassword>
    </PrincipalAuthenticationMechanism>

    <Authenticator>
      <AuthenticatorSequence>
        <SharedSecretChallengeResponse
method="urn:oasis:names:tc:SAML:2.0:ac:classes:Kerberos"/>
      </AuthenticatorSequence>
    </Authenticator>

  </AuthnMethod>

</AuthenticationContextDeclaration>
```

ПРИМЕЧАНИЕ 2. – Применение SSL описано в Приложении IV.

#### 12.3.4.4 MobileOneFactorUnregistered

**URI**: `urn:oasis:names:tc:SAML:2.0:ac:classes:MobileOneFactorUnregistered`

Этот URI используется также в качестве целевой области имен в схеме соответствующего класса контекста аутентификации в Приложении A.

Не отражает процедур регистрации мобильного пользователя и аутентификации мобильного устройства без явного взаимодействия с оконечным пользователем. Этот класс контекста аутентифицирует только устройство и никогда пользователя; он используется, когда службы, не являющиеся мобильным оператором, желают добавить в свой процесс аутентификации безопасную аутентификацию устройства.

```
<?xml version="1.0" encoding="UTF-8"?>

<xs:schema

targetNamespace="urn:oasis:names:tc:SAML:2.0:ac:classes:MobileOneFactorUnreg
istered"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns="urn:oasis:names:tc:SAML:2.0:ac:classes:MobileOneFactorUnregistered"
  finalDefault="extension"
  blockDefault="substitution"
  version="2.0">

  <xs:redefine schemaLocation="saml-schema-authn-context-types-2.0.xsd">
```

```
    <xs:annotation>
      <xs:documentation>
        Class identifier:
urn:oasis:names:tc:SAML:2.0:ac:classes:MobileOneFactorUnregistered
        Document identifier: saml-schema-authn-context-mobileonefactor-
unreg-2.0
        Location: http://docs.oasis-open.org/security/saml/v2.0/
        Revision history:
          V2.0 (March, 2005):
            New authentication context class schema for SAML V2.0.
      </xs:documentation>
    </xs:annotation>

    <xs:complexType name="AuthnContextDeclarationBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthnContextDeclarationBaseType">
          <xs:sequence>
            <xs:element ref="Identification" minOccurs="0"/>
            <xs:element ref="TechnicalProtection" minOccurs="0"/>
            <xs:element ref="OperationalProtection" minOccurs="0"/>
            <xs:element ref="AuthnMethod"/>
            <xs:element ref="GoverningAgreements" minOccurs="0"/>
            <xs:element ref="Extension" minOccurs="0"
maxOccurs="unbounded"/>
          </xs:sequence>
          <xs:attribute name="ID" type="xs:ID" use="optional"/>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="AuthnMethodBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthnMethodBaseType">
          <xs:sequence>
            <xs:element ref="PrincipalAuthenticationMechanism"
minOccurs="0"/>
            <xs:element ref="Authenticator"/>
            <xs:element ref="AuthenticatorTransportProtocol" minOccurs="0"/>
            <xs:element ref="Extension" minOccurs="0"
maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="AuthenticatorBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthenticatorBaseType">
          <xs:sequence>
            <xs:choice>
              <xs:element ref="DigSig"/>
              <xs:element ref="ZeroKnowledge"/>
              <xs:element ref="SharedSecretChallengeResponse"/>
              <xs:element ref="SharedSecretDynamicPlaintext"/>
              <xs:element ref="AsymmetricDecryption"/>
              <xs:element ref="AsymmetricKeyAgreement"/>
            </xs:choice>
            <xs:element ref="Extension" minOccurs="0"
maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="AuthenticatorTransportProtocolType">
      <xs:complexContent>
        <xs:restriction base="AuthenticatorTransportProtocolType">
          <xs:sequence>
            <xs:choice>
```

```
              <xs:element ref="SSL"/>
              <xs:element ref="MobileNetworkNoEncryption"/>
              <xs:element ref="MobileNetworkRadioEncryption"/>
              <xs:element ref="MobileNetworkEndToEndEncryption"/>
              <xs:element ref="WTLS"/>
            </xs:choice>
            <xs:element ref="Extension" minOccurs="0"
maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="OperationalProtectionType">
      <xs:complexContent>
        <xs:restriction base="OperationalProtectionType">
          <xs:sequence>
            <xs:element ref="SecurityAudit"/>
            <xs:element ref="DeactivationCallCenter"/>
            <xs:element ref="Extension" minOccurs="0"
maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="TechnicalProtectionBaseType">
      <xs:complexContent>
        <xs:restriction base="TechnicalProtectionBaseType">
          <xs:sequence>
            <xs:choice>
              <xs:element ref="PrivateKeyProtection"/>
              <xs:element ref="SecretKeyProtection"/>
            </xs:choice>
            <xs:element ref="Extension" minOccurs="0"
maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="PrivateKeyProtectionType">
      <xs:complexContent>
        <xs:restriction base="PrivateKeyProtectionType">
          <xs:sequence>
            <xs:element ref="KeyStorage"/>
            <xs:element ref="Extension" minOccurs="0"
maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="SecretKeyProtectionType">
      <xs:complexContent>
        <xs:restriction base="SecretKeyProtectionType">
          <xs:sequence>
            <xs:element ref="KeyStorage"/>
            <xs:element ref="Extension" minOccurs="0"
maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="KeyStorageType">
      <xs:complexContent>
        <xs:restriction base="KeyStorageType">
          <xs:attribute name="medium" use="required">
            <xs:simpleType>
              <xs:restriction base="mediumType">
```

```
                <xs:enumeration value="MobileDevice"/>
                <xs:enumeration value="MobileAuthCard"/>
                <xs:enumeration value="smartcard"/>
              </xs:restriction>
            </xs:simpleType>
          </xs:attribute>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="SecurityAuditType">
      <xs:complexContent>
        <xs:restriction base="SecurityAuditType">
          <xs:sequence>
            <xs:element ref="SwitchAudit"/>
            <xs:element ref="Extension" minOccurs="0"
maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="IdentificationType">
      <xs:complexContent>
        <xs:restriction base="IdentificationType">
          <xs:sequence>
            <xs:element ref="GoverningAgreements"/>
            <xs:element ref="Extension" minOccurs="0"
maxOccurs="unbounded"/>
          </xs:sequence>
          <xs:attribute name="nym">
            <xs:simpleType>
              <xs:restriction base="nymType">
                <xs:enumeration value="anonymity"/>
                <xs:enumeration value="pseudonymity"/>
              </xs:restriction>
            </xs:simpleType>
          </xs:attribute>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

  </xs:redefine>

</xs:schema>
```

ПРИМЕЧАНИЕ. – Применение SSL описано в Приложении IV.

#### 12.3.4.5 MobileTwoFactorUnregistered

**URI**: `urn:oasis:names:tc:SAML:2.0:ac:classes:MobileTwoFactorUnregistered`

Этот URI используется также в качестве целевой области имен в схеме соответствующего класса контекста аутентификации в Приложении A.

Не отражает процедур регистрации мобильного пользователя и аутентификации на базе двух факторов, таких как безопасное устройство и PIN пользователя. Этот класс контекста используется, когда служба, не являющаяся мобильным оператором, желает связать ID своего пользователя с услугой аутентификации на базе двух факторов, предоставляемой мобильным абонентам, используя в качестве регистрационных данных данные мобильного телефона.

```
<?xml version="1.0" encoding="UTF-8"?>

<xs:schema
targetNamespace="urn:oasis:names:tc:SAML:2.0:ac:classes:MobileTwoFactorUnreg
istered"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns="urn:oasis:names:tc:SAML:2.0:ac:classes:MobileTwoFactorUnregistered"
  finalDefault="extension"
  blockDefault="substitution"
  version="2.0">

  <xs:redefine schemaLocation="saml-schema-authn-context-types-2.0.xsd">
```

```
    <xs:annotation>
      <xs:documentation>
        Class identifier:
urn:oasis:names:tc:SAML:2.0:ac:classes:MobileTwoFactorUnregistered
        Document identifier: saml-schema-authn-context-mobiletwofactor-
unreg-2.0
        Location: http://docs.oasis-open.org/security/saml/v2.0/
        Revision history:
          V2.0 (March, 2005):
            New authentication context class schema for SAML V2.0.
      </xs:documentation>
    </xs:annotation>

    <xs:complexType name="AuthnContextDeclarationBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthnContextDeclarationBaseType">
          <xs:sequence>
            <xs:element ref="Identification" minOccurs="0"/>
            <xs:element ref="TechnicalProtection" minOccurs="0"/>
            <xs:element ref="OperationalProtection" minOccurs="0"/>
            <xs:element ref="AuthnMethod"/>
            <xs:element ref="GoverningAgreements" minOccurs="0"/>
            <xs:element ref="Extension" minOccurs="0"
maxOccurs="unbounded"/>
          </xs:sequence>
          <xs:attribute name="ID" type="xs:ID" use="optional"/>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="AuthnMethodBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthnMethodBaseType">
          <xs:sequence>
            <xs:element ref="PrincipalAuthenticationMechanism"
minOccurs="0"/>
            <xs:element ref="Authenticator"/>
            <xs:element ref="AuthenticatorTransportProtocol" minOccurs="0"/>
            <xs:element ref="Extension" minOccurs="0"
maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="AuthenticatorBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthenticatorBaseType">
          <xs:sequence>
            <xs:choice>
              <xs:element ref="DigSig"/>
              <xs:element ref="ZeroKnowledge"/>
              <xs:element ref="SharedSecretChallengeResponse"/>
              <xs:element ref="SharedSecretDynamicPlaintext"/>
              <xs:element ref="AsymmetricDecryption"/>
              <xs:element ref="AsymmetricKeyAgreement"/>
              <xs:element ref="ComplexAuthenticator"/>
            </xs:choice>
            <xs:element ref="Extension" minOccurs="0"
maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="ComplexAuthenticatorType">
      <xs:complexContent>
        <xs:restriction base="ComplexAuthenticatorType">
          <xs:sequence>
```

```
            <xs:choice>
              <xs:element ref="SharedSecretChallengeResponse"/>
              <xs:element ref="SharedSecretDynamicPlaintext"/>
            </xs:choice>
            <xs:element ref="Password"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="AuthenticatorTransportProtocolType">
      <xs:complexContent>
        <xs:restriction base="AuthenticatorTransportProtocolType">
          <xs:sequence>
            <xs:choice>
              <xs:element ref="SSL"/>
              <xs:element ref="MobileNetworkNoEncryption"/>
              <xs:element ref="MobileNetworkRadioEncryption"/>
              <xs:element ref="MobileNetworkEndToEndEncryption"/>
              <xs:element ref="WTLS"/>
            </xs:choice>
            <xs:element ref="Extension" minOccurs="0"
maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="OperationalProtectionType">
      <xs:complexContent>
        <xs:restriction base="OperationalProtectionType">
          <xs:sequence>
            <xs:element ref="SecurityAudit"/>
            <xs:element ref="DeactivationCallCenter"/>
            <xs:element ref="Extension" minOccurs="0"
maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="TechnicalProtectionBaseType">
      <xs:complexContent>
        <xs:restriction base="TechnicalProtectionBaseType">
          <xs:sequence>
            <xs:choice>
              <xs:element ref="PrivateKeyProtection"/>
              <xs:element ref="SecretKeyProtection"/>
            </xs:choice>
            <xs:element ref="Extension" minOccurs="0"
maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="PrivateKeyProtectionType">
      <xs:complexContent>
        <xs:restriction base="PrivateKeyProtectionType">
          <xs:sequence>
            <xs:element ref="KeyActivation"/>
            <xs:element ref="KeyStorage"/>
            <xs:element ref="Extension" minOccurs="0"
maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="SecretKeyProtectionType">
      <xs:complexContent>
```

```
        <xs:restriction base="SecretKeyProtectionType">
          <xs:sequence>
            <xs:element ref="KeyActivation"/>
            <xs:element ref="KeyStorage"/>
            <xs:element ref="Extension" minOccurs="0"
maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="KeyStorageType">
      <xs:complexContent>
        <xs:restriction base="KeyStorageType">
          <xs:attribute name="medium" use="required">
            <xs:simpleType>
              <xs:restriction base="mediumType">
                <xs:enumeration value="MobileDevice"/>
                <xs:enumeration value="MobileAuthCard"/>
                <xs:enumeration value="smartcard"/>
              </xs:restriction>
            </xs:simpleType>
          </xs:attribute>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="SecurityAuditType">
      <xs:complexContent>
        <xs:restriction base="SecurityAuditType">
          <xs:sequence>
            <xs:element ref="SwitchAudit"/>
            <xs:element ref="Extension" minOccurs="0"
maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="IdentificationType">
      <xs:complexContent>
        <xs:restriction base="IdentificationType">
          <xs:sequence>
            <xs:element ref="GoverningAgreements"/>
            <xs:element ref="Extension" minOccurs="0"
maxOccurs="unbounded"/>
          </xs:sequence>
          <xs:attribute name="nym">
            <xs:simpleType>
              <xs:restriction base="nymType">
                <xs:enumeration value="anonymity"/>
                <xs:enumeration value="pseudonymity"/>
              </xs:restriction>
            </xs:simpleType>
          </xs:attribute>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

  </xs:redefine>

</xs:schema>
```

#### 12.3.4.6 MobileOneFactorContract

**URI**: `urn:oasis:names:tc:SAML:2.0:ac:classes:MobileOneFactorContract`

Этот URI используется также в качестве целевой области имен в схеме соответствующего класса контекста аутентификации в Приложении А.

Отражает процедуры регистрации контракта мобильного пользователя и аутентификации на базе одного фактора. Например, устройства выполнения цифровой подписи с устройством хранения ключей, препятствующим их подделке, таким как номер мобильного абонента сети ISDN (MSISDN), но не требующей данных PIN или биометрических данных для аутентификации пользователя в реальном времени.

```
<?xml version="1.0" encoding="UTF-8"?>

<xs:schema

targetNamespace="urn:oasis:names:tc:SAML:2.0:ac:classes:MobileOneFactorContract"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns="urn:oasis:names:tc:SAML:2.0:ac:classes:MobileOneFactorContract"
  finalDefault="extension"
  blockDefault="substitution"
  version="2.0">

  <xs:redefine schemaLocation="saml-schema-authn-context-types-2.0.xsd">

    <xs:annotation>
      <xs:documentation>
        Class identifier:
urn:oasis:names:tc:SAML:2.0:ac:classes:MobileOneFactorContract
        Document identifier: saml-schema-authn-context-mobileonefactor-reg-2.0
        Location: http://docs.oasis-open.org/security/saml/v2.0/
        Revision history:
          V2.0 (March, 2005):
            New authentication context class schema for SAML V2.0.
      </xs:documentation>
    </xs:annotation>

    <xs:complexType name="AuthnContextDeclarationBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthnContextDeclarationBaseType">
          <xs:sequence>
            <xs:element ref="Identification" minOccurs="0"/>
            <xs:element ref="TechnicalProtection" minOccurs="0"/>
            <xs:element ref="OperationalProtection" minOccurs="0"/>
            <xs:element ref="AuthnMethod"/>
            <xs:element ref="GoverningAgreements" minOccurs="0"/>
            <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
          </xs:sequence>
          <xs:attribute name="ID" type="xs:ID" use="optional"/>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="AuthnMethodBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthnMethodBaseType">
          <xs:sequence>
            <xs:element ref="PrincipalAuthenticationMechanism" minOccurs="0"/>
            <xs:element ref="Authenticator"/>
            <xs:element ref="AuthenticatorTransportProtocol" minOccurs="0"/>
            <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="AuthenticatorBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthenticatorBaseType">
          <xs:sequence>
            <xs:choice>
              <xs:element ref="DigSig"/>
```

```
            <xs:element ref="ZeroKnowledge"/>
            <xs:element ref="SharedSecretChallengeResponse"/>
            <xs:element ref="SharedSecretDynamicPlaintext"/>
            <xs:element ref="AsymmetricDecryption"/>
            <xs:element ref="AsymmetricKeyAgreement"/>
          </xs:choice>
          <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
        </xs:sequence>
      </xs:restriction>
    </xs:complexContent>
  </xs:complexType>

  <xs:complexType name="AuthenticatorTransportProtocolType">
    <xs:complexContent>
      <xs:restriction base="AuthenticatorTransportProtocolType">
        <xs:sequence>
          <xs:choice>
            <xs:element ref="SSL"/>
            <xs:element ref="MobileNetworkNoEncryption"/>
            <xs:element ref="MobileNetworkRadioEncryption"/>
            <xs:element ref="MobileNetworkEndToEndEncryption"/>
            <xs:element ref="WTLS"/>
          </xs:choice>
          <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
        </xs:sequence>
      </xs:restriction>
    </xs:complexContent>
  </xs:complexType>

  <xs:complexType name="OperationalProtectionType">
    <xs:complexContent>
      <xs:restriction base="OperationalProtectionType">
        <xs:sequence>
          <xs:element ref="SecurityAudit"/>
          <xs:element ref="DeactivationCallCenter"/>
          <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
        </xs:sequence>
      </xs:restriction>
    </xs:complexContent>
  </xs:complexType>

  <xs:complexType name="TechnicalProtectionBaseType">
    <xs:complexContent>
      <xs:restriction base="TechnicalProtectionBaseType">
        <xs:sequence>
          <xs:choice>
            <xs:element ref="PrivateKeyProtection"/>
            <xs:element ref="SecretKeyProtection"/>
          </xs:choice>
          <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
        </xs:sequence>
      </xs:restriction>
    </xs:complexContent>
  </xs:complexType>

  <xs:complexType name="PrivateKeyProtectionType">
    <xs:complexContent>
      <xs:restriction base="PrivateKeyProtectionType">
        <xs:sequence>
          <xs:element ref="KeyStorage"/>
          <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
        </xs:sequence>
      </xs:restriction>
    </xs:complexContent>
  </xs:complexType>
```

```
    <xs:complexType name="SecretKeyProtectionType">
      <xs:complexContent>
        <xs:restriction base="SecretKeyProtectionType">
          <xs:sequence>
            <xs:element ref="KeyStorage"/>
            <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="KeyStorageType">
      <xs:complexContent>
        <xs:restriction base="KeyStorageType">
          <xs:attribute name="medium" use="required">
            <xs:simpleType>
              <xs:restriction base="mediumType">
                <xs:enumeration value="smartcard"/>
                <xs:enumeration value="MobileDevice"/>
                <xs:enumeration value="MobileAuthCard"/>
              </xs:restriction>
            </xs:simpleType>
          </xs:attribute>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="SecurityAuditType">
      <xs:complexContent>
        <xs:restriction base="SecurityAuditType">
          <xs:sequence>
            <xs:element ref="SwitchAudit"/>
            <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="IdentificationType">
      <xs:complexContent>
        <xs:restriction base="IdentificationType">
          <xs:sequence>
            <xs:element ref="PhysicalVerification"/>
            <xs:element ref="WrittenConsent"/>
            <xs:element ref="GoverningAgreements"/>
            <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
          </xs:sequence>
          <xs:attribute name="nym">
            <xs:simpleType>
              <xs:restriction base="nymType">
                <xs:enumeration value="anonymity"/>
                <xs:enumeration value="verinymity"/>
                <xs:enumeration value="pseudonymity"/>
              </xs:restriction>
            </xs:simpleType>
          </xs:attribute>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

  </xs:redefine>

</xs:schema>
```

#### 12.3.4.7 MobileTwoFactorContract

**URI**: `urn:oasis:names:tc:SAML:2.0:ac:classes:MobileTwoFactorContract`

Этот URI используется также в качестве целевой области имен в схеме соответствующего класса контекста аутентификации в Приложении А.

Отражает процедуры регистрации контракта мобильного пользователя и аутентификации на базе двух факторов. Например, устройства выполнения цифровой подписи с устройством хранения ключей, препятствующим их подделке, такие как SIM-карта GSM, которое требует явного подтверждения идентификации и намерений пользователя, например при помощи PIN или биометрических данных.

```
<?xml version="1.0" encoding="UTF-8"?>

<xs:schema
targetNamespace="urn:oasis:names:tc:SAML:2.0:ac:classes:MobileTwoFactorContr
act"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns="urn:oasis:names:tc:SAML:2.0:ac:classes:MobileTwoFactorContract"
  finalDefault="extension"
  blockDefault="substitution"
  version="2.0">

  <xs:redefine schemaLocation="saml-schema-authn-context-types-2.0.xsd">

    <xs:annotation>
      <xs:documentation>
        Class identifier:
urn:oasis:names:tc:SAML:2.0:ac:classes:MobileTwoFactorContract
        Document identifier: saml-schema-authn-context-mobiletwofactor-reg-
2.0
        Location: http://docs.oasis-open.org/security/saml/v2.0/
        Revision history:
          V2.0 (March, 2005):
            New authentication context class schema for SAML V2.0.
      </xs:documentation>
    </xs:annotation>

    <xs:complexType name="AuthnContextDeclarationBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthnContextDeclarationBaseType">
          <xs:sequence>
            <xs:element ref="Identification" minOccurs="0"/>
            <xs:element ref="TechnicalProtection" minOccurs="0"/>
            <xs:element ref="OperationalProtection" minOccurs="0"/>
            <xs:element ref="AuthnMethod"/>
            <xs:element ref="GoverningAgreements" minOccurs="0"/>
            <xs:element ref="Extension" minOccurs="0"
maxOccurs="unbounded"/>
          </xs:sequence>
          <xs:attribute name="ID" type="xs:ID" use="optional"/>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="AuthnMethodBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthnMethodBaseType">
          <xs:sequence>
            <xs:element ref="PrincipalAuthenticationMechanism"
minOccurs="0"/>
            <xs:element ref="Authenticator"/>
            <xs:element ref="AuthenticatorTransportProtocol" minOccurs="0"/>
            <xs:element ref="Extension" minOccurs="0"
maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="AuthenticatorBaseType">
      <xs:complexContent>
```

```
        <xs:restriction base="AuthenticatorBaseType">
          <xs:sequence>
            <xs:choice>
              <xs:element ref="DigSig"/>
              <xs:element ref="ZeroKnowledge"/>
              <xs:element ref="SharedSecretChallengeResponse"/>
              <xs:element ref="SharedSecretDynamicPlaintext"/>
              <xs:element ref="AsymmetricDecryption"/>
              <xs:element ref="AsymmetricKeyAgreement"/>
              <xs:element ref="ComplexAuthenticator"/>
            </xs:choice>
            <xs:element ref="Extension" minOccurs="0"
maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="ComplexAuthenticatorType">
      <xs:complexContent>
        <xs:restriction base="ComplexAuthenticatorType">
          <xs:sequence>
            <xs:choice>
              <xs:element ref="SharedSecretChallengeResponse"/>
              <xs:element ref="SharedSecretDynamicPlaintext"/>
            </xs:choice>
            <xs:element ref="Password"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="AuthenticatorTransportProtocolType">
      <xs:complexContent>
        <xs:restriction base="AuthenticatorTransportProtocolType">
          <xs:sequence>
            <xs:choice>
              <xs:element ref="SSL"/>
              <xs:element ref="MobileNetworkNoEncryption"/>
              <xs:element ref="MobileNetworkRadioEncryption"/>
              <xs:element ref="MobileNetworkEndToEndEncryption"/>
              <xs:element ref="WTLS"/>
            </xs:choice>
            <xs:element ref="Extension" minOccurs="0"
maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="OperationalProtectionType">
      <xs:complexContent>
        <xs:restriction base="OperationalProtectionType">
          <xs:sequence>
            <xs:element ref="SecurityAudit"/>
            <xs:element ref="DeactivationCallCenter"/>
            <xs:element ref="Extension" minOccurs="0"
maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="TechnicalProtectionBaseType">
      <xs:complexContent>
        <xs:restriction base="TechnicalProtectionBaseType">
          <xs:sequence>
            <xs:choice>
              <xs:element ref="PrivateKeyProtection"/>
              <xs:element ref="SecretKeyProtection"/>
            </xs:choice>
```

```
            <xs:element ref="Extension" minOccurs="0"
maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="PrivateKeyProtectionType">
      <xs:complexContent>
        <xs:restriction base="PrivateKeyProtectionType">
          <xs:sequence>
            <xs:element ref="KeyActivation"/>
            <xs:element ref="KeyStorage"/>
            <xs:element ref="Extension" minOccurs="0"
maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="SecretKeyProtectionType">
      <xs:complexContent>
        <xs:restriction base="SecretKeyProtectionType">
          <xs:sequence>
            <xs:element ref="KeyActivation"/>
            <xs:element ref="KeyStorage"/>
            <xs:element ref="Extension" minOccurs="0"
maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="KeyStorageType">
      <xs:complexContent>
        <xs:restriction base="KeyStorageType">
          <xs:attribute name="medium" use="required">
            <xs:simpleType>
              <xs:restriction base="mediumType">
                <xs:enumeration value="MobileDevice"/>
                <xs:enumeration value="MobileAuthCard"/>
                <xs:enumeration value="smartcard"/>
              </xs:restriction>
            </xs:simpleType>
          </xs:attribute>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="SecurityAuditType">
      <xs:complexContent>
        <xs:restriction base="SecurityAuditType">
          <xs:sequence>
            <xs:element ref="SwitchAudit"/>
            <xs:element ref="Extension" minOccurs="0"
maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="IdentificationType">
      <xs:complexContent>
        <xs:restriction base="IdentificationType">
          <xs:sequence>
            <xs:element ref="PhysicalVerification"/>
            <xs:element ref="WrittenConsent"/>
            <xs:element ref="GoverningAgreements"/>
            <xs:element ref="Extension" minOccurs="0"
maxOccurs="unbounded"/>
          </xs:sequence>
```

```
          <xs:attribute name="nym">
            <xs:simpleType>
              <xs:restriction base="nymType">
                <xs:enumeration value="anonymity"/>
                <xs:enumeration value="verinymity"/>
                <xs:enumeration value="pseudonymity"/>
              </xs:restriction>
            </xs:simpleType>
          </xs:attribute>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

  </xs:redefine>

</xs:schema>
```

ПРИМЕЧАНИЕ. – Применение SSL описано в Приложении IV.

#### 12.3.4.8 Password (пароль)

**URI**: `urn:oasis:names:tc:SAML:2.0:ac:classes:Password`

Этот URI используется также в качестве целевой области имен в схеме соответствующего класса контекста аутентификации в Приложении A.

Класс Password применяется, когда клиент аутентифицирует себя для органа по аутентификации путем передачи пароля в ходе незащищенного сеанса связи HTTP.

```
<?xml version="1.0" encoding="UTF-8"?>

<xs:schema targetNamespace="urn:oasis:names:tc:SAML:2.0:ac:classes:Password"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns="urn:oasis:names:tc:SAML:2.0:ac:classes:Password"
  finalDefault="extension"
  blockDefault="substitution"
  version="2.0">

  <xs:redefine schemaLocation="saml-schema-authn-context-types-2.0.xsd">

    <xs:annotation>
      <xs:documentation>
        Class identifier: urn:oasis:names:tc:SAML:2.0:ac:classes:Password
        Document identifier: saml-schema-authn-context-pword-2.0
        Location: http://docs.oasis-open.org/security/saml/v2.0/
        Revision history:
          V2.0 (March, 2005):
            New authentication context class schema for SAML V2.0.
      </xs:documentation>
    </xs:annotation>

    <xs:complexType name="AuthnContextDeclarationBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthnContextDeclarationBaseType">
          <xs:sequence>
            <xs:element ref="Identification" minOccurs="0"/>
            <xs:element ref="TechnicalProtection" minOccurs="0"/>
            <xs:element ref="OperationalProtection" minOccurs="0"/>
            <xs:element ref="AuthnMethod"/>
            <xs:element ref="GoverningAgreements" minOccurs="0"/>
            <xs:element ref="Extension" minOccurs="0"
maxOccurs="unbounded"/>
          </xs:sequence>
          <xs:attribute name="ID" type="xs:ID" use="optional"/>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="AuthnMethodBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthnMethodBaseType">
```

```
          <xs:sequence>
            <xs:element ref="PrincipalAuthenticationMechanism"
minOccurs="0"/>
            <xs:element ref="Authenticator"/>
            <xs:element ref="AuthenticatorTransportProtocol" minOccurs="0"/>
            <xs:element ref="Extension" minOccurs="0"
maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="AuthenticatorBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthenticatorBaseType">
          <xs:sequence>
            <xs:element ref="RestrictedPassword"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

  </xs:redefine>

</xs:schema>
```

Далее приведен пример элемента XML, соответствующего схеме этого класса контекста:

```
<AuthenticationContextDeclaration
  xmlns="urn:oasis:names:tc:SAML:2.0:ac:classes:Password">

  <AuthnMethod>
    <Authenticator>
      <AuthenticatorSequence>
        <RestrictedPassword>
          <Length min="4"/>
        </RestrictedPassword>
      </AuthenticatorSequence>
    </Authenticator>
  </AuthnMethod>

</AuthenticationContextDeclaration>
```

#### 12.3.4.9 PasswordProtectedTransport

**URI**: `urn:oasis:names:tc:SAML:2.0:ac:classes:PasswordProtectedTransport`

Этот URI используется также в качестве целевой области имен в схеме соответствующего класса контекста аутентификации в Приложении А.

Класс PasswordProtectedTransport применяется, когда клиент аутентифицирует себя для органа по аутентификации путем передачи пароля в ходе защищенного сеанса связи.

```
<?xml version="1.0" encoding="UTF-8"?>

<xs:schema
targetNamespace="urn:oasis:names:tc:SAML:2.0:ac:classes:PasswordProtectedTra
nsport"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns="urn:oasis:names:tc:SAML:2.0:ac:classes:PasswordProtectedTransport"
  finalDefault="extension"
  blockDefault="substitution"
  version="2.0">

  <xs:redefine schemaLocation="saml-schema-authn-context-types-2.0.xsd">

    <xs:annotation>
      <xs:documentation>
        Class identifier:
urn:oasis:names:tc:SAML:2.0:ac:classes:PasswordProtectedTransport
        Document identifier: saml-schema-authn-context-ppt-2.0
```

```
        Location: http://docs.oasis-open.org/security/saml/v2.0/
        Revision history:
          V2.0 (March, 2005):
            New authentication context class schema for SAML V2.0.
      </xs:documentation>
    </xs:annotation>

    <xs:complexType name="AuthnContextDeclarationBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthnContextDeclarationBaseType">
          <xs:sequence>
            <xs:element ref="Identification" minOccurs="0"/>
            <xs:element ref="TechnicalProtection" minOccurs="0"/>
            <xs:element ref="OperationalProtection" minOccurs="0"/>
            <xs:element ref="AuthnMethod"/>
            <xs:element ref="GoverningAgreements" minOccurs="0"/>
            <xs:element ref="Extension" minOccurs="0"
maxOccurs="unbounded"/>
          </xs:sequence>
          <xs:attribute name="ID" type="xs:ID" use="optional"/>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="AuthnMethodBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthnMethodBaseType">
          <xs:sequence>
            <xs:element ref="PrincipalAuthenticationMechanism"
minOccurs="0"/>
            <xs:element ref="Authenticator"/>
            <xs:element ref="AuthenticatorTransportProtocol"/>
            <xs:element ref="Extension" minOccurs="0"
maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="AuthenticatorBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthenticatorBaseType">
          <xs:sequence>
            <xs:element ref="RestrictedPassword"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="AuthenticatorTransportProtocolType">
      <xs:complexContent>
        <xs:restriction base="AuthenticatorTransportProtocolType">
          <xs:sequence>
            <xs:choice>
              <xs:element ref="SSL"/>
              <xs:element ref="MobileNetworkRadioEncryption"/>
              <xs:element ref="MobileNetworkEndToEndEncryption"/>
              <xs:element ref="WTLS"/>
              <xs:element ref="IPSec"/>
            </xs:choice>
            <xs:element ref="Extension" minOccurs="0"
maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

  </xs:redefine>

</xs:schema>
```

ПРИМЕЧАНИЕ. – Применение SSL описано в Приложении IV.

#### 12.3.4.10 PreviousSession

**URI**: `urn:oasis:names:tc:SAML:2.0:ac:classes:PreviousSession`

Этот URI используется также в качестве целевой области имен в схеме соответствующего класса контекста аутентификации в Приложении А.

Класс PreviousSession применяется, когда клиент уже однажды в прошлом аутентифицировал себя для органа по аутентификации, используя любой контекст аутентификации, поддерживаемый этим органом по аутентификации. Следовательно, последующее событие аутентификации, в ходе которого орган по аутентификации передаст сообщение доверяющей стороне, может быть значительно разнесено по времени от текущего запроса доступа к ресурсу от данного клиента.

Контекст аутентифицированного ранее сеанса связи в этот класс контекста явно не включается, потому что пользователь в течение этого сеанса связи не аутентифицируется, и, следовательно, механизм, использованный пользователем для аутентификации в предыдущем сеансе связи, не должен использоваться как часть принятия решения о том, разрешить ли пользователю доступ к этому ресурсу.

```
<?xml version="1.0" encoding="UTF-8"?>

<xs:schema
targetNamespace="urn:oasis:names:tc:SAML:2.0:ac:classes:PreviousSession"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns="urn:oasis:names:tc:SAML:2.0:ac:classes:PreviousSession"
  finalDefault="extension"
  blockDefault="substitution"
  version="2.0">

  <xs:redefine schemaLocation="saml-schema-authn-context-types-2.0.xsd">

    <xs:annotation>
      <xs:documentation>
        Class identifier:
urn:oasis:names:tc:SAML:2.0:ac:classes:PreviousSession
        Document identifier: saml-schema-authn-context-session-2.0
        Location: http://docs.oasis-open.org/security/saml/v2.0/
        Revision history:
          V2.0 (March, 2005):
            New authentication context class schema for SAML V2.0.
      </xs:documentation>
    </xs:annotation>

    <xs:complexType name="AuthnContextDeclarationBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthnContextDeclarationBaseType">
          <xs:sequence>
            <xs:element ref="Identification" minOccurs="0"/>
            <xs:element ref="TechnicalProtection" minOccurs="0"/>
            <xs:element ref="OperationalProtection" minOccurs="0"/>
            <xs:element ref="AuthnMethod"/>
            <xs:element ref="GoverningAgreements" minOccurs="0"/>
            <xs:element ref="Extension" minOccurs="0"
maxOccurs="unbounded"/>
          </xs:sequence>
          <xs:attribute name="ID" type="xs:ID" use="optional"/>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="AuthnMethodBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthnMethodBaseType">
          <xs:sequence>
            <xs:element ref="PrincipalAuthenticationMechanism"
minOccurs="0"/>
            <xs:element ref="Authenticator"/>
            <xs:element ref="AuthenticatorTransportProtocol" minOccurs="0"/>
            <xs:element ref="Extension" minOccurs="0"
maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>
```

```
    <xs:complexType name="AuthenticatorBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthenticatorBaseType">
          <xs:sequence>
            <xs:element ref="PreviousSession"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

  </xs:redefine>

</xs:schema>
```

#### 12.3.4.11 Открытый ключ – X.509

**URI**: `urn:oasis:names:tc:SAML:2.0:ac:classes:X509`

Этот URI используется также в качестве целевой области имен в схеме соответствующего класса контекста аутентификации в Приложении A.

Класс контекста X509 указывает, что клиент аутентифицирован при помощи цифровой подписи, ключ для которой был удостоверен как часть инфраструктуры открытых ключей X.509.

```
<?xml version="1.0" encoding="UTF-8"?>

<xs:schema targetNamespace="urn:oasis:names:tc:SAML:2.0:ac:classes:X509"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns="urn:oasis:names:tc:SAML:2.0:ac:classes:X509"
  finalDefault="extension"
  blockDefault="substitution"
  version="2.0">

  <xs:redefine schemaLocation="saml-schema-authn-context-types-2.0.xsd">

    <xs:annotation>
      <xs:documentation>
        Class identifier: urn:oasis:names:tc:SAML:2.0:ac:classes:X509
        Document identifier: saml-schema-authn-context-x509-2.0
        Location: http://docs.oasis-open.org/security/saml/v2.0/
        Revision history:
          V2.0 (March, 2005):
            New authentication context class schema for SAML V2.0.
      </xs:documentation>
    </xs:annotation>

    <xs:complexType name="AuthnContextDeclarationBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthnContextDeclarationBaseType">
          <xs:sequence>
            <xs:element ref="Identification" minOccurs="0"/>
            <xs:element ref="TechnicalProtection" minOccurs="0"/>
            <xs:element ref="OperationalProtection" minOccurs="0"/>
            <xs:element ref="AuthnMethod"/>
            <xs:element ref="GoverningAgreements" minOccurs="0"/>
            <xs:element ref="Extension" minOccurs="0"
maxOccurs="unbounded"/>
          </xs:sequence>
          <xs:attribute name="ID" type="xs:ID" use="optional"/>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="AuthnMethodBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthnMethodBaseType">
          <xs:sequence>
            <xs:element ref="PrincipalAuthenticationMechanism"/>
            <xs:element ref="Authenticator"/>
```

```
            <xs:element ref="AuthenticatorTransportProtocol" minOccurs="0"/>
            <xs:element ref="Extension" minOccurs="0"
maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="PrincipalAuthenticationMechanismType">
      <xs:complexContent>
        <xs:restriction base="PrincipalAuthenticationMechanismType">
          <xs:sequence>
            <xs:element ref="RestrictedPassword"/>
          </xs:sequence>
          <xs:attribute name="preauth" type="xs:integer" use="optional"/>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="AuthenticatorBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthenticatorBaseType">
          <xs:sequence>
            <xs:element ref="DigSig"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="PublicKeyType">
      <xs:complexContent>
        <xs:restriction base="PublicKeyType">
          <xs:attribute name="keyValidation" type="xs:anyURI"
fixed="urn:oasis:names:tc:SAML:2.0:ac:classes:X509"/>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

  </xs:redefine>

</xs:schema>
```

#### 12.3.4.12 Открытый ключ – PGP

**URI**: `urn:oasis:names:tc:SAML:2.0:ac:classes:PGP`

Этот URI используется также в качестве целевой области имен в схеме соответствующего класса контекста аутентификации в Приложении А.

Класс контекста PGP указывает, что клиент аутентифицирован при помощи цифровой подписи, ключ для которой был удостоверен как часть инфраструктуры открытых ключей PGP.

```
<?xml version="1.0" encoding="UTF-8"?>

<xs:schema targetNamespace="urn:oasis:names:tc:SAML:2.0:ac:classes:PGP"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns="urn:oasis:names:tc:SAML:2.0:ac:classes:PGP"
  finalDefault="extension"
  blockDefault="substitution"
  version="2.0">

  <xs:redefine schemaLocation="saml-schema-authn-context-types-2.0.xsd">

    <xs:annotation>
      <xs:documentation>
        Class identifier: urn:oasis:names:tc:SAML:2.0:ac:classes:PGP
        Document identifier: saml-schema-authn-context-pgp-2.0
        Location: http://docs.oasis-open.org/security/saml/v2.0/
        Revision history:
          V2.0 (March, 2005):
            New authentication context class schema for SAML V2.0.
```

```
      </xs:documentation>
    </xs:annotation>

    <xs:complexType name="AuthnContextDeclarationBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthnContextDeclarationBaseType">
          <xs:sequence>
            <xs:element ref="Identification" minOccurs="0"/>
            <xs:element ref="TechnicalProtection" minOccurs="0"/>
            <xs:element ref="OperationalProtection" minOccurs="0"/>
            <xs:element ref="AuthnMethod"/>
            <xs:element ref="GoverningAgreements" minOccurs="0"/>
            <xs:element ref="Extension" minOccurs="0"
maxOccurs="unbounded"/>
          </xs:sequence>
          <xs:attribute name="ID" type="xs:ID" use="optional"/>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="AuthnMethodBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthnMethodBaseType">
          <xs:sequence>
            <xs:element ref="PrincipalAuthenticationMechanism"/>
            <xs:element ref="Authenticator"/>
            <xs:element ref="AuthenticatorTransportProtocol" minOccurs="0"/>
            <xs:element ref="Extension" minOccurs="0"
maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="PrincipalAuthenticationMechanismType">
      <xs:complexContent>
        <xs:restriction base="PrincipalAuthenticationMechanismType">
          <xs:sequence>
            <xs:element ref="RestrictedPassword"/>
          </xs:sequence>
          <xs:attribute name="preauth" type="xs:integer" use="optional"/>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="AuthenticatorBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthenticatorBaseType">
          <xs:sequence>
            <xs:element ref="DigSig"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="PublicKeyType">
      <xs:complexContent>
        <xs:restriction base="PublicKeyType">
          <xs:attribute name="keyValidation"
fixed="urn:oasis:names:tc:SAML:2.0:ac:classes:PGP"/>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

  </xs:redefine>

</xs:schema>
```

#### 12.3.4.13 Открытый ключ – SPKI

**URI**: `urn:oasis:names:tc:SAML:2.0:ac:classes:SPKI`

Этот URI используется также в качестве целевой области имен в схеме соответствующего класса контекста аутентификации в Приложении А.

Класс контекста SPKI указывает, что клиент аутентифицирован при помощи цифровой подписи, ключ для которой был удостоверен как часть инфраструктуры открытых ключей SPKI.

```
<?xml version="1.0" encoding="UTF-8"?>

<xs:schema targetNamespace="urn:oasis:names:tc:SAML:2.0:ac:classes:SPKI"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns="urn:oasis:names:tc:SAML:2.0:ac:classes:SPKI"
  finalDefault="extension"
  blockDefault="substitution"
  version="2.0">

  <xs:redefine schemaLocation="saml-schema-authn-context-types-2.0.xsd">

    <xs:annotation>
      <xs:documentation>
        Class identifier: urn:oasis:names:tc:SAML:2.0:ac:classes:SPKI
        Document identifier: saml-schema-authn-context-spki-2.0
        Location: http://docs.oasis-open.org/security/saml/v2.0/
        Revision history:
          V2.0 (March, 2005):
            New authentication context class schema for SAML V2.0.
      </xs:documentation>
    </xs:annotation>

    <xs:complexType name="AuthnContextDeclarationBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthnContextDeclarationBaseType">
          <xs:sequence>
            <xs:element ref="Identification" minOccurs="0"/>
            <xs:element ref="TechnicalProtection" minOccurs="0"/>
            <xs:element ref="OperationalProtection" minOccurs="0"/>
            <xs:element ref="AuthnMethod"/>
            <xs:element ref="GoverningAgreements" minOccurs="0"/>
            <xs:element ref="Extension" minOccurs="0"
maxOccurs="unbounded"/>
          </xs:sequence>
          <xs:attribute name="ID" type="xs:ID" use="optional"/>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="AuthnMethodBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthnMethodBaseType">
          <xs:sequence>
            <xs:element ref="PrincipalAuthenticationMechanism"/>
            <xs:element ref="Authenticator"/>
            <xs:element ref="AuthenticatorTransportProtocol" minOccurs="0"/>
            <xs:element ref="Extension" minOccurs="0"
maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="PrincipalAuthenticationMechanismType">
      <xs:complexContent>
        <xs:restriction base="PrincipalAuthenticationMechanismType">
          <xs:sequence>
            <xs:element ref="RestrictedPassword"/>
          </xs:sequence>
          <xs:attribute name="preauth" type="xs:integer" use="optional"/>
        </xs:restriction>
```

```
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="AuthenticatorBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthenticatorBaseType">
          <xs:sequence>
            <xs:element ref="DigSig"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="PublicKeyType">
      <xs:complexContent>
        <xs:restriction base="PublicKeyType">
          <xs:attribute name="keyValidation"
fixed="urn:oasis:names:tc:SAML:2.0:ac:classes:SPKI"/>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

  </xs:redefine>

</xs:schema>
```

#### 12.3.4.14 Открытый ключ – цифровая подпись XML

**URI**: `urn:oasis:names:tc:SAML:2.0:ac:classes:XMLDSig`

Этот URI используется также в качестве целевой области имен в схеме соответствующего класса контекста аутентификации в Приложении А.

Этот класс контекста указывает, что клиент аутентифицирован при помощи цифровой подписи в соответствии с правилами, определенными в Правилах подписи XML Консорциума W3C.

```
<?xml version="1.0" encoding="UTF-8"?>

<xs:schema targetNamespace="urn:oasis:names:tc:SAML:2.0:ac:classes:XMLDSig"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns="urn:oasis:names:tc:SAML:2.0:ac:classes:XMLDSig"
  finalDefault="extension"
  blockDefault="substitution"
  version="2.0">

  <xs:redefine schemaLocation="saml-schema-authn-context-types-2.0.xsd">

    <xs:annotation>
      <xs:documentation>
        Class identifier: urn:oasis:names:tc:SAML:2.0:ac:classes:XMLDSig
        Document identifier: saml-schema-authn-context-xmldsig-2.0
        Location: http://docs.oasis-open.org/security/saml/v2.0/
        Revision history:
          V2.0 (March, 2005):
            New authentication context class schema for SAML V2.0.
      </xs:documentation>
    </xs:annotation>

    <xs:complexType name="AuthnContextDeclarationBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthnContextDeclarationBaseType">
          <xs:sequence>
            <xs:element ref="Identification" minOccurs="0"/>
            <xs:element ref="TechnicalProtection" minOccurs="0"/>
            <xs:element ref="OperationalProtection" minOccurs="0"/>
            <xs:element ref="AuthnMethod"/>
            <xs:element ref="GoverningAgreements" minOccurs="0"/>
            <xs:element ref="Extension" minOccurs="0"
maxOccurs="unbounded"/>
          </xs:sequence>
          <xs:attribute name="ID" type="xs:ID" use="optional"/>
        </xs:restriction>
```

```
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="AuthnMethodBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthnMethodBaseType">
          <xs:sequence>
            <xs:element ref="PrincipalAuthenticationMechanism"/>
            <xs:element ref="Authenticator"/>
            <xs:element ref="AuthenticatorTransportProtocol" minOccurs="0"/>
            <xs:element ref="Extension" minOccurs="0"
maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="PrincipalAuthenticationMechanismType">
      <xs:complexContent>
        <xs:restriction base="PrincipalAuthenticationMechanismType">
          <xs:sequence>
            <xs:element ref="RestrictedPassword"/>
          </xs:sequence>
          <xs:attribute name="preauth" type="xs:integer" use="optional"/>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="AuthenticatorBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthenticatorBaseType">
          <xs:sequence>
            <xs:element ref="DigSig"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="PublicKeyType">
      <xs:complexContent>
        <xs:restriction base="PublicKeyType">
          <xs:attribute name="keyValidation" type="xs:anyURI"
fixed="urn:ietf:rfc:3075"/>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

  </xs:redefine>

</xs:schema>
```

#### 12.3.4.15 Smartcard

**URI**: `urn:oasis:names:tc:SAML:2.0:ac:classes:Smartcard`

Этот URI используется также в качестве целевой области имен в схеме соответствующего класса контекста аутентификации в Приложении А.

Классом Smartcard указывается, когда клиент аутентифицирует себя для органа по аутентификации при помощи смарт-карты.

```
<?xml version="1.0" encoding="UTF-8"?>

<xs:schema
targetNamespace="urn:oasis:names:tc:SAML:2.0:ac:classes:Smartcard"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns="urn:oasis:names:tc:SAML:2.0:ac:classes:Smartcard"
  finalDefault="extension"
  blockDefault="substitution"
  version="2.0">

  <xs:redefine schemaLocation="saml-schema-authn-context-types-2.0.xsd">
```

```
    <xs:annotation>
      <xs:documentation>
        Class identifier: urn:oasis:names:tc:SAML:2.0:ac:classes:Smartcard
        Document identifier: saml-schema-authn-context-smartcard-2.0
        Location: http://docs.oasis-open.org/security/saml/v2.0/
        Revision history:
          V2.0 (March, 2005):
            New authentication context class schema for SAML V2.0.
      </xs:documentation>
    </xs:annotation>

    <xs:complexType name="AuthnContextDeclarationBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthnContextDeclarationBaseType">
          <xs:sequence>
            <xs:element ref="Identification" minOccurs="0"/>
            <xs:element ref="TechnicalProtection" minOccurs="0"/>
            <xs:element ref="OperationalProtection" minOccurs="0"/>
            <xs:element ref="AuthnMethod"/>
            <xs:element ref="GoverningAgreements" minOccurs="0"/>
            <xs:element ref="Extension" minOccurs="0"
maxOccurs="unbounded"/>
          </xs:sequence>
          <xs:attribute name="ID" type="xs:ID" use="optional"/>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="AuthnMethodBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthnMethodBaseType">
          <xs:sequence>
            <xs:element ref="PrincipalAuthenticationMechanism"/>
            <xs:element ref="Authenticator"/>
            <xs:element ref="AuthenticatorTransportProtocol" minOccurs="0"/>
            <xs:element ref="Extension" minOccurs="0"
maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="PrincipalAuthenticationMechanismType">
      <xs:complexContent>
        <xs:restriction base="PrincipalAuthenticationMechanismType">
          <xs:sequence>
            <xs:element ref="Smartcard"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

  </xs:redefine>

</xs:schema>
```

#### 12.3.4.16 SmartcardPKI

**URI**: `urn:oasis:names:tc:SAML:2.0:ac:classes:SmartcardPKI`

Этот URI используется также в качестве целевой области имен в схеме соответствующего класса контекста аутентификации в Приложении А.

Класс SmartcardPKI применяется, когда клиент аутентифицирует для органа по аутентификации при помощи механизма аутентификации, основанного на двух факторах, используя смарт-карту с вложенным в нее секретным ключом и PIN-код.

```
<?xml version="1.0" encoding="UTF-8"?>

<xs:schema
targetNamespace="urn:oasis:names:tc:SAML:2.0:ac:classes:SmartcardPKI"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns="urn:oasis:names:tc:SAML:2.0:ac:classes:SmartcardPKI"
  finalDefault="extension"
  blockDefault="substitution"
  version="2.0">

  <xs:redefine schemaLocation="saml-schema-authn-context-types-2.0.xsd">

    <xs:annotation>
      <xs:documentation>
        Class identifier:
urn:oasis:names:tc:SAML:2.0:ac:classes:SmartcardPKI
        Document identifier: saml-schema-authn-context-smartcardpki-2.0
        Location: http://docs.oasis-open.org/security/saml/v2.0/
        Revision history:
          V2.0 (March, 2005):
            New authentication context class schema for SAML V2.0.
      </xs:documentation>
    </xs:annotation>

    <xs:complexType name="AuthnContextDeclarationBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthnContextDeclarationBaseType">
          <xs:sequence>
            <xs:element ref="Identification" minOccurs="0"/>
            <xs:element ref="TechnicalProtection"/>
            <xs:element ref="OperationalProtection" minOccurs="0"/>
            <xs:element ref="AuthnMethod"/>
            <xs:element ref="GoverningAgreements" minOccurs="0"/>
            <xs:element ref="Extension" minOccurs="0"
maxOccurs="unbounded"/>
          </xs:sequence>
          <xs:attribute name="ID" type="xs:ID" use="optional"/>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="AuthnMethodBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthnMethodBaseType">
          <xs:sequence>
            <xs:element ref="PrincipalAuthenticationMechanism"/>
            <xs:element ref="Authenticator"/>
            <xs:element ref="AuthenticatorTransportProtocol" minOccurs="0"/>
            <xs:element ref="Extension" minOccurs="0"
maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="TechnicalProtectionBaseType">
      <xs:complexContent>
        <xs:restriction base="TechnicalProtectionBaseType">
          <xs:sequence>
            <xs:choice>
              <xs:element ref="PrivateKeyProtection"/>
            </xs:choice>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="PrincipalAuthenticationMechanismType">
      <xs:complexContent>
        <xs:restriction base="PrincipalAuthenticationMechanismType">
```

```
          <xs:sequence>
            <xs:element ref="Smartcard"/>
            <xs:element ref="ActivationPin"/>
            <xs:element ref="Extension" minOccurs="0"
maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="AuthenticatorBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthenticatorBaseType">
          <xs:sequence>
            <xs:choice>
              <xs:element ref="DigSig"/>
              <xs:element ref="AsymmetricDecryption"/>
              <xs:element ref="AsymmetricKeyAgreement"/>
            </xs:choice>
            <xs:element ref="Extension" minOccurs="0"
maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="PrivateKeyProtectionType">
      <xs:complexContent>
        <xs:restriction base="PrivateKeyProtectionType">
          <xs:sequence>
            <xs:element ref="KeyActivation"/>
            <xs:element ref="KeyStorage"/>
            <xs:element ref="Extension" minOccurs="0"
maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="KeyActivationType">
      <xs:complexContent>
        <xs:restriction base="KeyActivationType">
          <xs:sequence>
            <xs:element ref="ActivationPin"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="KeyStorageType">
      <xs:complexContent>
        <xs:restriction base="KeyStorageType">
          <xs:attribute name="medium" use="required">
            <xs:simpleType>
              <xs:restriction base="mediumType">
                <xs:enumeration value="smartcard"/>
              </xs:restriction>
            </xs:simpleType>
          </xs:attribute>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

  </xs:redefine>

</xs:schema>
```

#### 12.3.4.17 SoftwarePKI

**URI**: urn:oasis:names:tc:SAML:2.0:ac:classes:SoftwarePKI

Этот URI используется также в качестве целевой области имен в схеме соответствующего класса контекста аутентификации в Приложении А.

Класс SoftwarePKI применяется, когда клиент использует сертификат X.509, сохраненный в программном виде, с целью аутентификации себя для органа по аутентификации.

```
<?xml version="1.0" encoding="UTF-8"?>

<xs:schema
targetNamespace="urn:oasis:names:tc:SAML:2.0:ac:classes:SoftwarePKI"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns="urn:oasis:names:tc:SAML:2.0:ac:classes:SoftwarePKI"
  finalDefault="extension"
  blockDefault="substitution"
  version="2.0">

  <xs:redefine schemaLocation="saml-schema-authn-context-types-2.0.xsd">

    <xs:annotation>
      <xs:documentation>
        Class identifier: urn:oasis:names:tc:SAML:2.0:ac:classes:SoftwarePKI
        Document identifier: saml-schema-authn-context-softwarepki-2.0
        Location: http://docs.oasis-open.org/security/saml/v2.0/
        Revision history:
          V2.0 (March, 2005):
            New authentication context class schema for SAML V2.0.
      </xs:documentation>
    </xs:annotation>

    <xs:complexType name="AuthnContextDeclarationBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthnContextDeclarationBaseType">
          <xs:sequence>
            <xs:element ref="Identification" minOccurs="0"/>
            <xs:element ref="TechnicalProtection"/>
            <xs:element ref="OperationalProtection" minOccurs="0"/>
            <xs:element ref="AuthnMethod"/>
            <xs:element ref="GoverningAgreements" minOccurs="0"/>
            <xs:element ref="Extension" minOccurs="0"
maxOccurs="unbounded"/>
          </xs:sequence>
          <xs:attribute name="ID" type="xs:ID" use="optional"/>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="AuthnMethodBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthnMethodBaseType">
          <xs:sequence>
            <xs:element ref="PrincipalAuthenticationMechanism"/>
            <xs:element ref="Authenticator"/>
            <xs:element ref="AuthenticatorTransportProtocol" minOccurs="0"/>
            <xs:element ref="Extension" minOccurs="0"
maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="TechnicalProtectionBaseType">
      <xs:complexContent>
        <xs:restriction base="TechnicalProtectionBaseType">
          <xs:sequence>
            <xs:choice>
              <xs:element ref="PrivateKeyProtection"/>
            </xs:choice>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
```

```
    </xs:complexType>

    <xs:complexType name="PrincipalAuthenticationMechanismType">
      <xs:complexContent>
        <xs:restriction base="PrincipalAuthenticationMechanismType">
          <xs:sequence>
            <xs:element ref="ActivationPin"/>
            <xs:element ref="Extension" minOccurs="0"
maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="AuthenticatorBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthenticatorBaseType">
          <xs:sequence>
            <xs:choice>
              <xs:element ref="DigSig"/>
              <xs:element ref="AsymmetricDecryption"/>
              <xs:element ref="AsymmetricKeyAgreement"/>
            </xs:choice>
            <xs:element ref="Extension" minOccurs="0"
maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="PrivateKeyProtectionType">
      <xs:complexContent>
        <xs:restriction base="PrivateKeyProtectionType">
          <xs:sequence>
            <xs:element ref="KeyActivation"/>
            <xs:element ref="KeyStorage"/>
            <xs:element ref="Extension" minOccurs="0"
maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="KeyActivationType">
      <xs:complexContent>
        <xs:restriction base="KeyActivationType">
          <xs:sequence>
            <xs:element ref="ActivationPin"/>
            <xs:element ref="Extension" minOccurs="0"
maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="KeyStorageType">
      <xs:complexContent>
        <xs:restriction base="KeyStorageType">
          <xs:attribute name="medium" use="required">
            <xs:simpleType>
              <xs:restriction base="mediumType">
                <xs:enumeration value="memory"/>
              </xs:restriction>
            </xs:simpleType>
          </xs:attribute>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

  </xs:redefine>
</xs:schema>
```

#### 12.3.4.18 Телефония

**URI**: `urn:oasis:names:tc:SAML:2.0:ac:classes:Telephony`

Этот URI используется также в качестве целевой области имен в схеме соответствующего класса контекста аутентификации в Приложении А.

Этот класс используется для указания того, что клиент аутентифицирован при помощи представления фиксированного телефонного номера, переданного по протоколу телефонной связи, такой как ADSL.

```
<?xml version="1.0" encoding="UTF-8"?>

<xs:schema
targetNamespace="urn:oasis:names:tc:SAML:2.0:ac:classes:Telephony"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns="urn:oasis:names:tc:SAML:2.0:ac:classes:Telephony"
  finalDefault="extension"
  blockDefault="substitution"
  version="2.0">

  <xs:redefine schemaLocation="saml-schema-authn-context-types-2.0.xsd">

    <xs:annotation>
      <xs:documentation>
        Class identifier: urn:oasis:names:tc:SAML:2.0:ac:classes:Telephony
        Document identifier: saml-schema-authn-context-telephony-2.0
        Location: http://docs.oasis-open.org/security/saml/v2.0/
        Revision history:
          V2.0 (March, 2005):
            New authentication context class schema for SAML V2.0.
      </xs:documentation>
    </xs:annotation>

    <xs:complexType name="AuthnContextDeclarationBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthnContextDeclarationBaseType">
          <xs:sequence>
            <xs:element ref="Identification" minOccurs="0"/>
            <xs:element ref="TechnicalProtection" minOccurs="0"/>
            <xs:element ref="OperationalProtection" minOccurs="0"/>
            <xs:element ref="AuthnMethod"/>
            <xs:element ref="GoverningAgreements" minOccurs="0"/>
            <xs:element ref="Extension" minOccurs="0"
maxOccurs="unbounded"/>
          </xs:sequence>
          <xs:attribute name="ID" type="xs:ID" use="optional"/>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="AuthnMethodBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthnMethodBaseType">
          <xs:sequence>
            <xs:element ref="PrincipalAuthenticationMechanism"
minOccurs="0"/>
            <xs:element ref="Authenticator"/>
            <xs:element ref="AuthenticatorTransportProtocol"/>
            <xs:element ref="Extension" minOccurs="0"
maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="AuthenticatorBaseType">
      <xs:complexContent>
```

```
        <xs:restriction base="AuthenticatorBaseType">
          <xs:sequence>
            <xs:element ref="SubscriberLineNumber"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="AuthenticatorTransportProtocolType">
      <xs:complexContent>
        <xs:restriction base="AuthenticatorTransportProtocolType">
          <xs:sequence>
            <xs:choice>
              <xs:element ref="PSTN"/>
              <xs:element ref="ISDN"/>
              <xs:element ref="ADSL"/>
            </xs:choice>
            <xs:element ref="Extension" minOccurs="0"
maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

  </xs:redefine>

</xs:schema>
```

#### 12.3.4.19 Телефония (номадическая)

**URI**: `urn:oasis:names:tc:SAML:2.0:ac:classes:NomadTelephony`

Этот URI используется также в качестве целевой области имен в схеме соответствующего класса контекста аутентификации в Приложении А.

Указывает, что клиент "передвигается" (возможно, используя телефонную карту) и аутентифицирует себя при помощи номера линии, суффикса пользователя и пароля.

```
<?xml version="1.0" encoding="UTF-8"?>

<xs:schema
targetNamespace="urn:oasis:names:tc:SAML:2.0:ac:classes:NomadTelephony"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns="urn:oasis:names:tc:SAML:2.0:ac:classes:NomadTelephony"
  finalDefault="extension"
  blockDefault="substitution"
  version="2.0">

  <xs:redefine schemaLocation="saml-schema-authn-context-types-2.0.xsd">

    <xs:annotation>
      <xs:documentation>
        Class identifier:
urn:oasis:names:tc:SAML:2.0:ac:classes:NomadTelephony
        Document identifier: saml-schema-authn-context-nomad-telephony-2.0
        Location: http://docs.oasis-open.org/security/saml/v2.0/
        Revision history:
          V2.0 (March, 2005):
            New authentication context class schema for SAML V2.0.
      </xs:documentation>
    </xs:annotation>

    <xs:complexType name="AuthnContextDeclarationBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthnContextDeclarationBaseType">
          <xs:sequence>
            <xs:element ref="Identification" minOccurs="0"/>
            <xs:element ref="TechnicalProtection" minOccurs="0"/>
            <xs:element ref="OperationalProtection" minOccurs="0"/>
            <xs:element ref="AuthnMethod"/>
            <xs:element ref="GoverningAgreements" minOccurs="0"/>
```

```
            <xs:element ref="Extension" minOccurs="0"
maxOccurs="unbounded"/>
          </xs:sequence>
          <xs:attribute name="ID" type="xs:ID" use="optional"/>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="AuthnMethodBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthnMethodBaseType">
          <xs:sequence>
            <xs:element ref="PrincipalAuthenticationMechanism"
minOccurs="0"/>
            <xs:element ref="Authenticator"/>
            <xs:element ref="AuthenticatorTransportProtocol"/>
            <xs:element ref="Extension" minOccurs="0"
maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="AuthenticatorBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthenticatorBaseType">
          <xs:sequence>
            <xs:element ref="Password"/>
            <xs:element ref="SubscriberLineNumber"/>
            <xs:element ref="UserSuffix"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="AuthenticatorTransportProtocolType">
      <xs:complexContent>
        <xs:restriction base="AuthenticatorTransportProtocolType">
          <xs:sequence>
            <xs:choice>
              <xs:element ref="PSTN"/>
              <xs:element ref="ISDN"/>
              <xs:element ref="ADSL"/>
            </xs:choice>
            <xs:element ref="Extension" minOccurs="0"
maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

  </xs:redefine>

</xs:schema>
```

#### 12.3.4.20 Телефония (персональная)

**URI**: `urn:oasis:names:tc:SAML:2.0:ac:classes:PersonalTelephony`

Этот URI используется также в качестве целевой области имен в схеме соответствующего класса контекста аутентификации в Приложении А.

Этот класс используется для указания того, что клиент аутентифицирован при помощи представления фиксированного телефонного номера и суффикса пользователя, переданного по протоколу телефонной связи, такой как ADSL.

```
<?xml version="1.0" encoding="UTF-8"?>

<xs:schema
targetNamespace="urn:oasis:names:tc:SAML:2.0:ac:classes:PersonalizedTelephony"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns="urn:oasis:names:tc:SAML:2.0:ac:classes:PersonalizedTelephony"
```

```
  finalDefault="extension"
  blockDefault="substitution"
  version="2.0">

  <xs:redefine schemaLocation="saml-schema-authn-context-types-2.0.xsd">

    <xs:annotation>
      <xs:documentation>
        Class identifier:
urn:oasis:names:tc:SAML:2.0:ac:classes:PersonalizedTelephony
        Document identifier: saml-schema-authn-context-personal-telephony-2.0
        Location: http://docs.oasis-open.org/security/saml/v2.0/
        Revision history:
          V2.0 (March, 2005):
            New authentication context class schema for SAML V2.0.
      </xs:documentation>
    </xs:annotation>

    <xs:complexType name="AuthnContextDeclarationBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthnContextDeclarationBaseType">
          <xs:sequence>
            <xs:element ref="Identification" minOccurs="0"/>
            <xs:element ref="TechnicalProtection" minOccurs="0"/>
            <xs:element ref="OperationalProtection" minOccurs="0"/>
            <xs:element ref="AuthnMethod"/>
            <xs:element ref="GoverningAgreements" minOccurs="0"/>
            <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
          </xs:sequence>
          <xs:attribute name="ID" type="xs:ID" use="optional"/>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="AuthnMethodBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthnMethodBaseType">
          <xs:sequence>
            <xs:element ref="PrincipalAuthenticationMechanism" minOccurs="0"/>
            <xs:element ref="Authenticator"/>
            <xs:element ref="AuthenticatorTransportProtocol"/>
            <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="AuthenticatorBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthenticatorBaseType">
          <xs:sequence>
            <xs:element ref="SubscriberLineNumber"/>
            <xs:element ref="UserSuffix"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="AuthenticatorTransportProtocolType">
      <xs:complexContent>
        <xs:restriction base="AuthenticatorTransportProtocolType">
          <xs:sequence>
            <xs:choice>
              <xs:element ref="PSTN"/>
              <xs:element ref="ISDN"/>
              <xs:element ref="ADSL"/>
            </xs:choice>
            <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:restriction>
```

```
      </xs:complexContent>
    </xs:complexType>

  </xs:redefine>

</xs:schema>
```

#### 12.3.4.21 Телефония (аутентифицированная)

**URI**: `urn:oasis:names:tc:SAML:2.0:ac:classes:AuthenticatedTelephony`

Этот URI используется также в качестве целевой области имен в схеме соответствующего класса контекста аутентификации в Приложении А.

Указывает, что клиент аутентифицирует себя при помощи номера линии, суффикса пользователя и пароля.

```
<?xml version="1.0" encoding="UTF-8"?>

<xs:schema
targetNamespace="urn:oasis:names:tc:SAML:2.0:ac:classes:AuthenticatedTelepho
ny"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns="urn:oasis:names:tc:SAML:2.0:ac:classes:AuthenticatedTelephony"
  finalDefault="extension"
  blockDefault="substitution"
  version="2.0">

  <xs:redefine schemaLocation="saml-schema-authn-context-types-2.0.xsd">

    <xs:annotation>
      <xs:documentation>
        Class identifier:
urn:oasis:names:tc:SAML:2.0:ac:classes:AuthenticatedTelephony
        Document identifier: saml-schema-authn-context-auth-telephony-2.0
        Location: http://docs.oasis-open.org/security/saml/v2.0/
        Revision history:
          V2.0 (March, 2005):
            New authentication context class schema for SAML V2.0.
      </xs:documentation>
    </xs:annotation>

    <xs:complexType name="AuthnContextDeclarationBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthnContextDeclarationBaseType">
          <xs:sequence>
            <xs:element ref="Identification" minOccurs="0"/>
            <xs:element ref="TechnicalProtection" minOccurs="0"/>
            <xs:element ref="OperationalProtection" minOccurs="0"/>
            <xs:element ref="AuthnMethod"/>
            <xs:element ref="GoverningAgreements" minOccurs="0"/>
            <xs:element ref="Extension" minOccurs="0"
maxOccurs="unbounded"/>
          </xs:sequence>
          <xs:attribute name="ID" type="xs:ID" use="optional"/>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="AuthnMethodBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthnMethodBaseType">
          <xs:sequence>
            <xs:element ref="PrincipalAuthenticationMechanism"
minOccurs="0"/>
            <xs:element ref="Authenticator"/>
            <xs:element ref="AuthenticatorTransportProtocol"/>
            <xs:element ref="Extension" minOccurs="0"
maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>
```

```
    <xs:complexType name="AuthenticatorBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthenticatorBaseType">
          <xs:sequence>
            <xs:element ref="Password"/>
            <xs:element ref="SubscriberLineNumber"/>
            <xs:element ref="UserSuffix"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="AuthenticatorTransportProtocolType">
      <xs:complexContent>
        <xs:restriction base="AuthenticatorTransportProtocolType">
          <xs:sequence>
            <xs:choice>
              <xs:element ref="PSTN"/>
              <xs:element ref="ISDN"/>
              <xs:element ref="ADSL"/>
            </xs:choice>
            <xs:element ref="Extension" minOccurs="0"
maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>
  </xs:redefine>

</xs:schema>
```

#### 12.3.4.22 SecureRemotePassword

**URI**: `urn:oasis:names:tc:SAML:2.0:ac:classes:SecureRemotePassword`

Этот URI используется также в качестве целевой области имен в схеме соответствующего класса контекста аутентификации в Приложении А.

Класс SecureRemotePassword применяется, когда аутентификация выполняется при помощи безопасного удаленного пароля, как определено в IETF RFC 2945.

```
<?xml version="1.0" encoding="UTF-8"?>

<xs:schema
targetNamespace="urn:oasis:names:tc:SAML:2.0:ac:classes:SecureRemotePassword"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns="urn:oasis:names:tc:SAML:2.0:ac:classes:SecureRemotePassword"
  finalDefault="extension"
  blockDefault="substitution"
  version="2.0">

  <xs:redefine schemaLocation="saml-schema-authn-context-types-2.0.xsd">

    <xs:annotation>
      <xs:documentation>
        Class identifier: urn:oasis:names:tc:SAML:2.0:ac:classes:SecureRemotePassword
        Document identifier: saml-schema-authn-context-srp-2.0
        Location: http://docs.oasis-open.org/security/saml/v2.0/
        Revision history:
          V2.0 (March, 2005):
            New authentication context class schema for SAML V2.0.
      </xs:documentation>
    </xs:annotation>

    <xs:complexType name="AuthnContextDeclarationBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthnContextDeclarationBaseType">
          <xs:sequence>
            <xs:element ref="Identification" minOccurs="0"/>
            <xs:element ref="TechnicalProtection" minOccurs="0"/>
            <xs:element ref="OperationalProtection" minOccurs="0"/>
            <xs:element ref="AuthnMethod"/>
```

```
            <xs:element ref="GoverningAgreements" minOccurs="0"/>
            <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
          </xs:sequence>
          <xs:attribute name="ID" type="xs:ID" use="optional"/>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="AuthnMethodBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthnMethodBaseType">
          <xs:sequence>
            <xs:element ref="PrincipalAuthenticationMechanism"/>
            <xs:element ref="Authenticator"/>
            <xs:element ref="AuthenticatorTransportProtocol" minOccurs="0"/>
            <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="PrincipalAuthenticationMechanismType">
      <xs:complexContent>
        <xs:restriction base="PrincipalAuthenticationMechanismType">
          <xs:sequence>
            <xs:element ref="RestrictedPassword"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="AuthenticatorBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthenticatorBaseType">
          <xs:sequence>
            <xs:element ref="SharedSecretChallengeResponse"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="SharedSecretChallengeResponseType">
      <xs:complexContent>
        <xs:restriction base="SharedSecretChallengeResponseType">
          <xs:attribute name="method" type="xs:anyURI" fixed="urn:ietf:rfc:2945"/>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

  </xs:redefine>

</xs:schema>
```

#### 12.3.4.23 Аутентификация клиента TLS на базе сертификатов

**URI**: `urn:oasis:names:tc:SAML:2.0:ac:classes:TLSClient`

Этот URI используется также в качестве целевой области имен в схеме соответствующего класса контекста аутентификации в Приложении А.

Этот класс указывает, что клиент аутентифицирован при помощи сертификата клиента, безопасность которого обеспечена транспортировкой TLS.

```
<?xml version="1.0" encoding="UTF-8"?>

<xs:schema
targetNamespace="urn:oasis:names:tc:SAML:2.0:ac:classes:TLSClient"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns="urn:oasis:names:tc:SAML:2.0:ac:classes:TLSClient"
  finalDefault="extension"
  blockDefault="substitution"
  version="2.0">
```

```
  <xs:redefine schemaLocation="saml-schema-authn-context-types-2.0.xsd">

    <xs:annotation>
      <xs:documentation>
        Class identifier: urn:oasis:names:tc:SAML:2.0:ac:classes:TLSClient
        Document identifier: saml-schema-authn-context-sslcert-2.0
        Location: http://docs.oasis-open.org/security/saml/v2.0/
        Revision history:
          V2.0 (March, 2005):
            New authentication context class schema for SAML V2.0.
      </xs:documentation>
    </xs:annotation>

    <xs:complexType name="AuthnContextDeclarationBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthnContextDeclarationBaseType">
          <xs:sequence>
            <xs:element ref="Identification" minOccurs="0"/>
            <xs:element ref="TechnicalProtection" minOccurs="0"/>
            <xs:element ref="OperationalProtection" minOccurs="0"/>
            <xs:element ref="AuthnMethod"/>
            <xs:element ref="GoverningAgreements" minOccurs="0"/>
            <xs:element ref="Extension" minOccurs="0"
maxOccurs="unbounded"/>
          </xs:sequence>
          <xs:attribute name="ID" type="xs:ID" use="optional"/>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="AuthnMethodBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthnMethodBaseType">
          <xs:sequence>
            <xs:element ref="PrincipalAuthenticationMechanism"/>
            <xs:element ref="Authenticator"/>
            <xs:element ref="AuthenticatorTransportProtocol" minOccurs="0"/>
            <xs:element ref="Extension" minOccurs="0"
maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="PrincipalAuthenticationMechanismType">
      <xs:complexContent>
        <xs:restriction base="PrincipalAuthenticationMechanismType">
          <xs:sequence>
            <xs:element ref="RestrictedPassword"/>
          </xs:sequence>
          <xs:attribute name="preauth" type="xs:integer" use="optional"/>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="AuthenticatorBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthenticatorBaseType">
          <xs:sequence>
            <xs:element ref="DigSig"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="PublicKeyType">
      <xs:complexContent>
        <xs:restriction base="PublicKeyType">
          <xs:attribute name="keyValidation" type="xs:anyURI"
fixed="urn:oasis:names:tc:SAML:2.0:ac:classes:X509"/>
```

```
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="AuthenticatorTransportProtocolType">
      <xs:complexContent>
        <xs:restriction base="AuthenticatorTransportProtocolType">
          <xs:sequence>
            <xs:choice>
              <xs:element ref="SSL"/>
              <xs:element ref="WTLS"/>
            </xs:choice>
            <xs:element ref="Extension" minOccurs="0"
maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

  </xs:redefine>

</xs:schema>
```

ПРИМЕЧАНИЕ. – Применение SSL описано в Приложении IV.

#### 12.3.4.24 TimeSyncToken

**URI**: `urn:oasis:names:tc:SAML:2.0:ac:classes:TimeSyncToken`

Этот URI используется также в качестве целевой области имен в схеме соответствующего класса контекста аутентификации в Приложении A.

Класс TimeSyncToken применяется, когда клиент аутентифицируется при помощи метки времени синхронизации.

```
<?xml version="1.0" encoding="UTF-8"?>

<xs:schema
targetNamespace="urn:oasis:names:tc:SAML:2.0:ac:classes:TimeSyncToken"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns="urn:oasis:names:tc:SAML:2.0:ac:classes:TimeSyncToken"
  finalDefault="extension"
  blockDefault="substitution"
  version="2.0">

  <xs:redefine schemaLocation="saml-schema-authn-context-types-2.0.xsd">

    <xs:annotation>
      <xs:documentation>
        Class identifier:
urn:oasis:names:tc:SAML:2.0:ac:classes:TimeSyncToken
        Document identifier: saml-schema-authn-context-timesync-2.0
        Location: http://docs.oasis-open.org/security/saml/v2.0/
        Revision history:
          V2.0 (March, 2005):
            New authentication context class schema for SAML V2.0.
      </xs:documentation>
    </xs:annotation>

    <xs:complexType name="AuthnContextDeclarationBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthnContextDeclarationBaseType">
          <xs:sequence>
            <xs:element ref="Identification" minOccurs="0"/>
            <xs:element ref="TechnicalProtection" minOccurs="0"/>
            <xs:element ref="OperationalProtection" minOccurs="0"/>
            <xs:element ref="AuthnMethod"/>
            <xs:element ref="GoverningAgreements" minOccurs="0"/>
            <xs:element ref="Extension" minOccurs="0"
maxOccurs="unbounded"/>
          </xs:sequence>
          <xs:attribute name="ID" type="xs:ID" use="optional"/>
        </xs:restriction>
```

```
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="AuthnMethodBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthnMethodBaseType">
          <xs:sequence>
            <xs:element ref="PrincipalAuthenticationMechanism"
minOccurs="0"/>
            <xs:element ref="Authenticator"/>
            <xs:element ref="AuthenticatorTransportProtocol" minOccurs="0"/>
            <xs:element ref="Extension" minOccurs="0"
maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="PrincipalAuthenticationMechanismType">
      <xs:complexContent>
        <xs:restriction base="PrincipalAuthenticationMechanismType">
          <xs:sequence>
            <xs:element ref="Token"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="TokenType">
      <xs:complexContent>
        <xs:restriction base="TokenType">
          <xs:sequence>
            <xs:element ref="TimeSyncToken"/>
            <xs:element ref="Extension" minOccurs="0"
maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="TimeSyncTokenType">
      <xs:complexContent>
        <xs:restriction base="TimeSyncTokenType">
          <xs:attribute name="DeviceType" use="required">
            <xs:simpleType>
              <xs:restriction base="DeviceTypeType">
                <xs:enumeration value="hardware"/>
              </xs:restriction>
            </xs:simpleType>
          </xs:attribute>

          <xs:attribute name="SeedLength" use="required">
            <xs:simpleType>
              <xs:restriction base="xs:integer">
                <xs:minInclusive value="64"/>
              </xs:restriction>
            </xs:simpleType>
          </xs:attribute>

          <xs:attribute name="DeviceInHand" use="required">
            <xs:simpleType>
              <xs:restriction base="booleanType">
                <xs:enumeration value="true"/>
              </xs:restriction>
            </xs:simpleType>
          </xs:attribute>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

  </xs:redefine>

</xs:schema>
```

#### 12.3.4.25 Неопределенный

**URI**: `urn:oasis:names:tc:SAML:2.0:ac:classes:Unspecified`

Класс Неопределенный указывает, что аутентификация была выполнена при помощи неопределенных средств.

## 13 Требования по соответствию для языка SAML

В настоящем разделе описываются возможности, которые являются обязательными и дополнительными для вариантов реализации, претендующих на соответствие с SAML.

Настоящая Рекомендация определяет множество профилей, имеющих название. Каждый профиль (отличный от профилей атрибутов) описывает подробности выбранных потоков сообщений SAML и может рассматриваться как неразделимая функция, которая может быть реализована в виде программного компонента. Вариант реализации профиля предусматривает использование связи для каждого обмена сообщениями, включенного в профиль. Связь можно рассматривать как конкретный метод реализации для выполнения обмена сообщениями.

В настоящем разделе перечислены все различные профили, определенные в настоящей Рекомендации. Для каждого профиля перечислены соответствующие ему потоки сообщений SAML V2.0, и для каждого потока сообщений описывается множество возможных связей. Комбинация профиля, обмена сообщениями и выбранной связи называется *возможностью* SAML V2.0.

В настоящем разделе описывается также матрица соответствия для языка SAML V2.0. Определяется множество различных *режимов работы* или ролей. Матрица соответствия описывает набор возможностей, который должен быть реализован для каждого режима работы.

### 13.1 Профили SAML и возможный вариант реализации

В таблице 2 перечислены все профили, определенные профилями SAML. Для каждого профиля описан протокол потока сообщений, имеющийся в данном профиле. В финальном столбце для каждого потока сообщений приведен список приемлемых связей.

**Таблица 1/X.1141 – Возможные варианты реализации**

| Профиль | Потоки сообщений | Связь |
|---|---|---|
| SSO веб-браузера | `<AuthnRequest>` от SP к IdP | HTTP Redirect |
| | | HTTP POST |
| | | HTTP Artifact |
| | IdP `<Response>` к SP | HTTP POST |
| | | HTTP Artifact |
| Расширенная архитектура SSO "клиент/прокси-сервер" | ECP к SP, SP к ECP к IdP | PAOS |
| | IdP к ECP к SP, SP к ECP | PAOS |
| Обнаружение провайдера идентификации | Установка маркера HTTP | HTTP |
| | Установка маркера HTTP | HTTP |
| Единый выход из системы | `<LogoutRequest>` | HTTP Redirect |
| | | HTTP POST |
| | | HTTP Artifact |
| | | SOAP |
| | `<LogoutResponse>` | HTTP Redirect |
| | | HTTP POST |
| | | HTTP Artifact |
| | | SOAP |
| Управление идентификатором имени | `<ManageNameIDRequest>` | HTTP Redirect |
| | | HTTP POST |
| | | HTTP Artifact |
| | | SOAP |
| | `<ManageNameIDResponse>` | HTTP Redirect |
| | | SOAP |
| Разрешение артефакта | `<ArtifactResolve>`, `<ArtifactResponse>` | SOAP |

**Таблица 1/X.1141 – Возможные варианты реализации**

| Профиль | Потоки сообщений | Связь |
|---|---|---|
| Запрос аутентификации | <AuthnQuery>, <Response> | SOAP |
| Запрос атрибута | <AttributeQuery>, <Response> | SOAP |
| Запрос решения по авторизации | <AuthzDecisionQuery>, <Response> | SOAP |
| Запрос подтверждения идентификатора | <AssertionIDRequest>, <Response> | SOAP |
| Преобразование идентификатора имени | <NameIDMappingRequest>, <NameIDMappingResponse> | SOAP |
| Связь SAML URI | GET, HTTP Ответ | HTTP |
| Профиль атрибута UUID | | |
| Профиль атрибута DCE PAC | | |
| Профиль атрибута X.500 | | |
| Профиль атрибута XACML | | |
| Метаданные | Использование | |
| | Обмен | |

## 13.2 Соответствие

В настоящем разделе описываются технические требования по соответствию для языка SAML V2.0.

### 13.2.1 Режимы работы

В настоящей Рекомендации используется выражение "режим работы" для описания роли, которую может играть компонент программного обеспечения в обеспечении соответствия языку SAML. Режимы работы могу быть следующими:

- IdP – провайдер идентификации;
- IdP Lite – содержание провайдера идентификации;
- SP – провайдер услуг;
- SP Lite – содержание провайдера услуг;
- ECP – расширенная архитектура "клиент/прокси-сервер";
- ответственный орган SAML по атрибутам;
- ответственный орган SAML по авторизации;
- ответственный орган SAML по аутентификации;
- запрашивающая сторона SAML.

### 13.2.2 Матрица возможностей

Приведенные далее матрицы (таблица 2) определяют уникальные множества требований по соответствию при помощи трех компонентов, взятых из таблицы 1 в следующей форме: профиль, сообщение(я), связь. Компонент сообщение включается не всегда, если он очевиден из контекста.

**Таблица 2/X.1141 – Матрица возможностей**

| Возможность | IdP | IdP lite | SP | SP lite | ECP |
|---|---|---|---|---|---|
| Веб SSO, `<AuthnRequest>`, HTTP redirect | Должен | Должен | Должен | Должен | N/A |
| Веб SSO, `<Response>`, HTTP POST | Должен | Должен | Должен | Должен | N/A |
| Веб SSO, `<Response>`, HTTP artifact | Должен | Должен | Должен | Должен | N/A |
| Разрешение артефакта, SOAP | Должен | Должен | Должен | Должен | N/A |
| Расширенная архитектура "клиент/прокси-сервер"SSO, PAOS | Должен | Должен | Должен | Должен | Должен |
| Управление идентификатором имени, HTTP Redirect (инициирована IdP) | Должен | Не должен | Должен | Не должен | N/A |
| Управление идентификатором имени, SOAP (инициирована IdP) | Должен | Не должен | Дополн. | Не должен | N/A |
| Управление идентификатором имени, HTTP redirect<br>ПРИМЕЧАНИЕ (информативное). – PE11 (см. OASIS PE:2006) предлагает добавить (инициирована SP) | Должен | Не должен | Должен | Не должен | N/A |
| Управление идентификатором имени, SOAP (инициирована SP) | Должен | Не должен | Дополн. | Не должен | N/A |
| Единый выход из системы (инициирован IdP) – HTTP redirect | Должен | Должен | Должен | Должен | N/A |
| Единый выход из системы (инициирован IdP) – SOAP | Должен | Дополн. | Должен | Дополн. | N/A |
| Единый выход из системы (инициирован SP) – HTTP redirect | Должен | Должен | Должен | Должен | N/A |
| Единый выход из системы (инициирован SP) – SOAP | Должен | Дополн. | Должен | Дополн. | N/A |
| Обнаружение провайдера идентификации (HTTP маркер) | Должен | Должен | Дополн. | Дополн. | N/A |

ПРИМЕЧАНИЕ 1 (информативное). – PE16 (см. OASIS PE:2006) предлагает заменить N/A на "Дополнительный" в последней строке последнего столбца таблицы 2.

ПРИМЕЧАНИЕ 2 (информативное). – PE25 (см. OASIS PE:2006) предлагает добавить в конец таблицы 2 следующие строки:

| Возможность | IdP | IdP Lite | SP | SP Lite | ECP |
|---|---|---|---|---|---|
| Структуры метаданных | Дополн. | Дополн. | Дополн. | Дополн. | N/A |
| Взаимодействие метаданных | Дополн. | Дополн. | Дополн. | Дополн. | N/A |

ПРИМЕЧАНИЕ 3 (информативное). – PE29 (см. OASIS PE:2006) предлагает добавить в конец таблицы 2 следующие строки:

| Возможность | IdP | IdP Lite | SP | SP Lite | ECP |
|---|---|---|---|---|---|
| Запрос идентификатора подтверждения | Дополн. | N/A | N/A | N/A | N/A |
| Связь SAML URI | Дополн. | N/A | N/A | N/A | N/A |

В таблице 3 показаны режимы работы, которые расширяют определенные выше режимы IdP или SP. Их необходимо понимать как комбинацию режима IdP или SP из вышеприведенной таблицы с соответствующие множеством возможностей расширения, показанных далее.

**Таблица 3/X.1141 – Расширенные режимы IdP, SP**

| Возможность | Расширенный режим IdP | Расширенный режим SP |
|---|---|---|
| Прокси-элемент провайдера идентификации | Должен | Должен |
| Преобразование идентификатора имени, SOAP | Должен | Должен |

В таблице 4 показаны требования по соответствию Ответственных органов SAML и запрашивающих сторон.

**Таблица 4/X.1141 – Матрица ответственных органов SAML и запрашивающих сторон**

| Возможность | Орган аутентификации SAML | Орган атрибута SAML | Орган SAML принятия решения по авторизации | Запрашивающая сторона SAML |
|---|---|---|---|---|
| Запрос аутентификации, SOAP | Должен | Дополн. | Дополн. | Дополн. |
| Запрос атрибута, SOAP | Дополн. | Должен | Дополн. | Дополн. |
| Запрос решения по авторизации, SOAP | Дополн. | Дополн. | Должен | Дополн. |
| Запрос идентификатора подтверждения, SOAP | Должен | Должен | Должен | Дополн. |
| Связь SAML URI | Должен | Должен | Должен | Дополн. |

ПРИМЕЧАНИЕ 4 (информативное). – PE25 и PE42 (см. OASIS PE:2006) предлагают изменить таблицу 4 следующим образом:

| Возможность | Орган аутентификации SAML | Орган атрибута SAML | Орган SAML принятия решения по авторизации | Запрашивающая сторона SAML |
|---|---|---|---|---|
| Запрос аутентификации, SOAP | Должен | N/A | N/A | Дополн. |
| Запрос атрибута, SOAP | N/A | Должен | N/A | Дополн. |
| Запрос решения по авторизации, SOAP | N/A | N/A | Должен | Дополн. |
| Запрос идентификатора подтверждения, SOAP | Должен | Должен | Должен | Дополн. |
| Связь SAML URI | Должен | Должен | Должен | Дополн. |
| Структуры метаданных | Дополн. | Дополн. | Дополн. | Дополн. |
| Взаимодействие метаданных | Дополн. | Дополн. | Дополн. | Дополн. |

### 13.2.3 Реализация идентификаторов, определенных в языке SAML

Все используемые режимы работы должны использовать следующие идентификаторы, определенные в SAML:

- все идентификаторы формата атрибута имени, определенные в разделе 8;
- все идентификаторы формата имени идентификатора, определенные в разделе 8.

Обеспечение соответствия вариантам реализации SAML должно разрешать использование всех констант идентификатора (см. 8.1 и 8.2) в ходе создания и использования сообщении SAML. Создатели сообщений SAML должны иметь возможность создавать сообщения, а потребители сообщений SAML должны иметь возможность обрабатывать сообщения с любыми константами, определенными в настоящих разделах.

Постоянные идентификаторы имен и временные идентификаторы имен определяют нормативные правила обработки для создателя таких идентификаторов. Все нормативные правила обработки должны поддерживаться при помощи соответствующих вариантов реализации. Остальные идентификаторы не определяют нормативные правила обработки. Отсюда, создание и использование этих идентификаторов имеет смысл только, когда создающие и использующие стороны имеют внешне определенное соглашение об интерпретации семантики идентификаторов.

ПРИМЕЧАНИЕ. – В этом контексте, "обработать" означает, что вариант реализации должен успешно выполнить синтаксический разбор и обработать идентификатор без потерь или возвращения ошибки. Каким образом этот вариант реализации работает с идентификатором после того, как он обработан на этом уровне, выходит за рамки настоящей Рекомендации.

Реализация SAML может предоставить возможности, описанные выше в результате прямой поддержки идентификаторов или за счет использования поддерживаемых программных интерфейсов. Интерфейсы, предусмотренные для этой цели, должны позволять программное расширение реализации SAML для обработки всех идентификаторов, которые не обработаны самим вариантом реализации.

### 13.2.4 Исполнение шифрованных элементов

Все используемые режимы работы должны иметь возможность обрабатывать или создавать следующие шифрованные элементы в любом контексте, где требуется обрабатывать или создавать соответствующие шифрованные элементы, а именно `<saml:NameID>`, `<saml:Assertion>` или `<saml:Attribute>`:

- `<saml:EncryptedID>`;
- `<saml:EncryptedAssertion>`;
- `<saml:EncryptedAttribute>`.

### 13.2.5 Модели безопасности для связей SOAP и URI

Приведенные далее модели безопасности являются обязательными для реализации во всех профилях, реализованных с использованием связи SOAP, а также для связи SAML URI. Ответственные органы SAML и запрашивающие стороны должны реализовать следующие методы аутентификации:

- нет аутентификации клиента или сервера;
- базовая аутентификация HTTP с TLS 1.0 или без него. Запрашивающая сторона SAML должна обязательно передавать заголовок авторизации с исходным запросом;
- аутентификация сервера HTTP по TLS 1.0 с сертификатом на стороне сервера;
- взаимная аутентификация HTTP по TLS 1.0 с сертификатом, как на стороне сервера, так и на стороне клиента.

Если ответственный орган SAML использует TLS 1.0, он должен использовать сертификат на стороне сервера.

ПРИМЕЧАНИЕ 1 (информативное). – PE25 (см. OASIS PE:2006) предлагает добавить новый подраздел о структуре метаданных следующего содержания:

Варианты реализации, претендующие на соответствие SAML, могут объявлять соответствие каждого режима работы метаданным SAML при помощи выбора опции Структура метаданных. Относительно каждого режима работы, такое соответствие означает следующее:

Реализация метаданных SAML в соответствии с расширенным форматом метаданных SAML во всех случаях, когда взаимодействующий равноправный элемент имеет опцию, обозначенную в спецификации SAML, в зависимости от существования метаданных SAML. Выбор опции Структура метаданных влечет за собой требование, согласно которому такие метаданные должны быть доступны для взаимодействующего равноправного элемента. Возможность взаимодействия метаданных, описанная далее предоставляет средства для выполнения этого требования.

Ссылки, использование и точное выполнение метаданных SAML в соответствии со взаимодействующим равноправным элементом, когда известны метаданные, относящиеся к этому равноправному элементу и к определенному действию, и текущий обмен сообщениями завершен или более не может оставаться в кэше, при условии, что метаданные доступны и не запрещены правилами или определенным действием и данным конкретным обменом сообщениями.

ПРИМЕЧАНИЕ 2 (информативное). – PE25 (см. OASIS PE:2006) предлагает добавить новый подраздел о взаимодействии метаданных следующего содержания:

Выбор опции взаимодействия метаданных влечет за собой требование реализации, в дополнение ко всем другим механизмам, механизмов публикации данных о местоположении и разрешения, описанных в метаданных SAML в разделе 9.

## 13.3 Цифровая подпись XML и шифрование XML

Язык SAML V2.0 использует подпись XML для выполнения функций подписи XML и шифрования для обеспечения целостности и аутентификации источника. Язык SAML V2.0 использует шифрование XML для обеспечения конфиденциальности, включая шифрованные идентификаторы, шифрованные подтверждения и шифрованные атрибуты.

### 13.3.1 Алгоритмы подписи XML

Раздел 6.1 в правилах подписи XML Консорциума W3C требует обязательного использования следующего:

- профиля сообщения: SHA-1;
- MAC: HMAC-SHA1;
- канонического назначения каналов XML: CanonicalXML (без комментариев);
- Transform: подписи, запечатанной в конверт.

Следовательно, они должны быть реализованы при помощи вариантов, соответствующих SAML V2.0.

Кроме того, для обеспечения взаимодействия, при помощи вариантов, соответствующих SAML V2.0 должно быть реализовано следующее:

- подпись: DSAwithSHA1 (рекомендована в Правилах подписи W3C, необходима для взаимодействия).

Хотя подпись XML требует обязательного использования алгоритма подписи DSAwithSHA1, это не требуется в языке SAML V2.0, но рекомендуется.

ПРИМЕЧАНИЕ. – NIST (Национальный институт стандартов и технологии) сегодня приветствует использование SHA-256 (защищенный алгоритм хеширования с закодированными 256-битовыми ключами) вместо SHA-1.

### 13.3.2 Алгоритмы шифрования XML

- Разделы 5.2.1 и 5.2.2 Правил шифрования XML Консорциума W3C требует обязательного использования следующих алгоритмов: Блочное шифрование: Триада DES, AES-128, AES-256.
- Транспортировка ключа: RSA-v1.5, RSA-OAEP.

Следовательно, вышеупомянутые алгоритмы должны быть реализованы при помощи вариантов, соответствующих SAML V2.0.

### 13.4 Использование TLS 1.0

В любых вариантах SAML V2.0, использующих TLS 1.0, серверы должны аутентифицировать себя для клиентов, используя сертификат X.509 v3. Клиент должен установить идентификатор сервера на основе содержания сертификата (обычно путем проверки поля DN в объекте сертификата).

#### 13.4.1 Связь SAML SOAP и URI

Варианты реализации, имеющие возможности TLS, должны реализовывать модуль шифрования TLS_RSA_WITH_3DES_EDE_CBC_SHA и могут реализовывать модуль TLS_RSA_AES_128_CBC_SHA.

Варианты реализации FIPS, имеющие возможности TLS, должны реализовывать соответствующий модуль шифрования TLS_RSA_FIPS_WITH_3DES_EDE_CBC_SHA, и могут реализовывать соответствующий модуль шифрования TLS_RSA_FIPS_AES_128_CBC_SHA.

#### 13.4.2 Профили языка SAML для SSO веб-браузера

Варианты реализации, имеющие возможности TLS, должны реализовывать модуль шифрования TLS_RSA_WITH_3DES_EDE_CBC_SHA (см. IETF RFC 2246).

## Приложение A

## Схемы SAML

Настоящее приложение является составной частью настоящей Рекомендации. Оно содержит листинги требуемых Схем SAML.

### A.1 Схема SAML – Формулировка

Это листинг схемы SAML – Формулировка.

```
<?xml version="1.0" encoding="US-ASCII"?>
<schema
    targetNamespace="urn:oasis:names:tc:SAML:2.0:assertion"
    xmlns="http://www.w3.org/2001/XMLSchema"
    xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion"
    xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
    xmlns:xenc="http://www.w3.org/2001/04/xmlenc#"
    elementFormDefault="unqualified"
    attributeFormDefault="unqualified"
    blockDefault="substitution"
    version="2.0">
    <import namespace="http://www.w3.org/2000/09/xmldsig#"
        schemaLocation="http://www.w3.org/TR/2002/REC-xmldsig-core-
20020212/xmldsig-core-schema.xsd"/>
    <import namespace="http://www.w3.org/2001/04/xmlenc#"
        schemaLocation="http://www.w3.org/TR/2002/REC-xmlenc-core-20021210/xenc-
schema.xsd"/>
    <annotation>
        <documentation>
            Document identifier: saml-schema-assertion-2.0
            Location: http://docs.oasis-open.org/security/saml/v2.0/
            Revision history:
            V1.0 (November, 2002):
              Initial Standard Schema.
            V1.1 (September, 2003):
              Updates within the same V1.0 namespace.
            V2.0 (March, 2005):
              New assertion schema for SAML V2.0  namespace.
        </documentation>
    </annotation>
    <attributeGroup name="IDNameQualifiers">
        <attribute name="NameQualifier" type="string" use="optional"/>
        <attribute name="SPNameQualifier" type="string" use="optional"/>
    </attributeGroup>
    <element name="BaseID" type="saml:BaseIDAbstractType"/>
    <complexType name="BaseIDAbstractType" abstract="true">
        <attributeGroup ref="saml:IDNameQualifiers"/>
    </complexType>
    <element name="NameID" type="saml:NameIDType"/>
    <complexType name="NameIDType">
```

```
        <simpleContent>
            <extension base="string">
                <attributeGroup ref="saml:IDNameQualifiers"/>
                <attribute name="Format" type="anyURI" use="optional"/>
                <attribute name="SPProvidedID" type="string" use="optional"/>
            </extension>
        </simpleContent>
    </complexType>
    <complexType name="EncryptedElementType">
        <sequence>
            <element ref="xenc:EncryptedData"/>
            <element ref="xenc:EncryptedKey" minOccurs="0" maxOccurs="unbounded"/>
        </sequence>
    </complexType>
    <element name="EncryptedID" type="saml:EncryptedElementType"/>
    <element name="Issuer" type="saml:NameIDType"/>
    <element name="AssertionIDRef" type="NCName"/>
    <element name="AssertionURIRef" type="anyURI"/>
    <element name="Assertion" type="saml:AssertionType"/>
    <complexType name="AssertionType">
        <sequence>
            <element ref="saml:Issuer"/>
            <element ref="ds:Signature" minOccurs="0"/>
            <element ref="saml:Subject" minOccurs="0"/>
            <element ref="saml:Conditions" minOccurs="0"/>
            <element ref="saml:Advice" minOccurs="0"/>
            <choice minOccurs="0" maxOccurs="unbounded">
                <element ref="saml:Statement"/>
                <element ref="saml:AuthnStatement"/>
                <element ref="saml:AuthzDecisionStatement"/>
                <element ref="saml:AttributeStatement"/>
            </choice>
        </sequence>
        <attribute name="Version" type="string" use="required"/>
        <attribute name="ID" type="ID" use="required"/>
        <attribute name="IssueInstant" type="dateTime" use="required"/>
    </complexType>
    <element name="Subject" type="saml:SubjectType"/>
    <complexType name="SubjectType">
        <choice>
            <sequence>
                <choice>
                    <element ref="saml:BaseID"/>
                    <element ref="saml:NameID"/>
                    <element ref="saml:EncryptedID"/>
                </choice>
                <element ref="saml:SubjectConfirmation" minOccurs="0"
maxOccurs="unbounded"/>
            </sequence>
            <element ref="saml:SubjectConfirmation" maxOccurs="unbounded"/>
        </choice>
    </complexType>
    <element name="SubjectConfirmation" type="saml:SubjectConfirmationType"/>
    <complexType name="SubjectConfirmationType">
        <sequence>
            <choice minOccurs="0">
                <element ref="saml:BaseID"/>
                <element ref="saml:NameID"/>
                <element ref="saml:EncryptedID"/>
            </choice>
            <element ref="saml:SubjectConfirmationData" minOccurs="0"/>
        </sequence>
        <attribute name="Method" type="anyURI" use="required"/>
    </complexType>
    <element name="SubjectConfirmationData"
type="saml:SubjectConfirmationDataType"/>
    <complexType name="SubjectConfirmationDataType" mixed="true">
        <complexContent>
            <restriction base="anyType">
                <sequence>
```

```
                    <any namespace="##any" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
                </sequence>
                <attribute name="NotBefore" type="dateTime" use="optional"/>
                <attribute name="NotOnOrAfter" type="dateTime" use="optional"/>
                <attribute name="Recipient" type="anyURI" use="optional"/>
                <attribute name="InResponseTo" type="NCName" use="optional"/>
                <attribute name="Address" type="string" use="optional"/>
                <anyAttribute namespace="##other" processContents="lax"/>
            </restriction>
        </complexContent>
    </complexType>
    <complexType name="KeyInfoConfirmationDataType" mixed="false">
        <complexContent>
            <restriction base="saml:SubjectConfirmationDataType">
                <sequence>
                    <element ref="ds:KeyInfo" maxOccurs="unbounded"/>
                </sequence>
            </restriction>
        </complexContent>
    </complexType>
    <element name="Conditions" type="saml:ConditionsType"/>
    <complexType name="ConditionsType">
        <choice minOccurs="0" maxOccurs="unbounded">
            <element ref="saml:Condition"/>
            <element ref="saml:AudienceRestriction"/>
            <element ref="saml:OneTimeUse"/>
            <element ref="saml:ProxyRestriction"/>
        </choice>
        <attribute name="NotBefore" type="dateTime" use="optional"/>
        <attribute name="NotOnOrAfter" type="dateTime" use="optional"/>
    </complexType>
    <element name="Condition" type="saml:ConditionAbstractType"/>
    <complexType name="ConditionAbstractType" abstract="true"/>
    <element name="AudienceRestriction" type="saml:AudienceRestrictionType"/>
    <complexType name="AudienceRestrictionType">
        <complexContent>
            <extension base="saml:ConditionAbstractType">
                <sequence>
                    <element ref="saml:Audience" maxOccurs="unbounded"/>
                </sequence>
            </extension>
        </complexContent>
    </complexType>
    <element name="Audience" type="anyURI"/>
    <element name="OneTimeUse" type="saml:OneTimeUseType" />
    <complexType name="OneTimeUseType">
        <complexContent>
            <extension base="saml:ConditionAbstractType"/>
        </complexContent>
    </complexType>
    <element name="ProxyRestriction" type="saml:ProxyRestrictionType"/>
    <complexType name="ProxyRestrictionType">
    <complexContent>
        <extension base="saml:ConditionAbstractType">
            <sequence>
                <element ref="saml:Audience" minOccurs="0" maxOccurs="unbounded"/>
            </sequence>
            <attribute name="Count" type="nonNegativeInteger" use="optional"/>
        </extension>
 </complexContent>
    </complexType>
    <element name="Advice" type="saml:AdviceType"/>
    <complexType name="AdviceType">
        <choice minOccurs="0" maxOccurs="unbounded">
            <element ref="saml:AssertionIDRef"/>
            <element ref="saml:AssertionURIRef"/>
            <element ref="saml:Assertion"/>
            <element ref="saml:EncryptedAssertion"/>
            <any namespace="##other" processContents="lax"/>
        </choice>
```

```
    </complexType>
    <element name="EncryptedAssertion" type="saml:EncryptedElementType"/>
    <element name="Statement" type="saml:StatementAbstractType"/>
    <complexType name="StatementAbstractType" abstract="true"/>
    <element name="AuthnStatement" type="saml:AuthnStatementType"/>
    <complexType name="AuthnStatementType">
        <complexContent>
            <extension base="saml:StatementAbstractType">
                <sequence>
                    <element ref="saml:SubjectLocality" minOccurs="0"/>
                    <element ref="saml:AuthnContext"/>
                </sequence>
                <attribute name="AuthnInstant" type="dateTime" use="required"/>
                <attribute name="SessionIndex" type="string" use="optional"/>
                <attribute name="SessionNotOnOrAfter" type="dateTime"
use="optional"/>
            </extension>
        </complexContent>
    </complexType>
    <element name="SubjectLocality" type="saml:SubjectLocalityType"/>
    <complexType name="SubjectLocalityType">
        <attribute name="Address" type="string" use="optional"/>
        <attribute name="DNSName" type="string" use="optional"/>
    </complexType>
    <element name="AuthnContext" type="saml:AuthnContextType"/>
    <complexType name="AuthnContextType">
        <sequence>
            <choice>
                <sequence>
                    <element ref="saml:AuthnContextClassRef"/>
                    <choice minOccurs="0">
                        <element ref="saml:AuthnContextDecl"/>
                        <element ref="saml:AuthnContextDeclRef"/>
                    </choice>
                </sequence>
                <choice>
                    <element ref="saml:AuthnContextDecl"/>
                    <element ref="saml:AuthnContextDeclRef"/>
                </choice>
            </choice>
            <element ref="saml:AuthenticatingAuthority" minOccurs="0"
maxOccurs="unbounded"/>
        </sequence>
    </complexType>
    <element name="AuthnContextClassRef" type="anyURI"/>
    <element name="AuthnContextDeclRef" type="anyURI"/>
    <element name="AuthnContextDecl" type="anyType"/>
    <element name="AuthenticatingAuthority" type="anyURI"/>
    <element name="AuthzDecisionStatement"
type="saml:AuthzDecisionStatementType"/>
    <complexType name="AuthzDecisionStatementType">
        <complexContent>
            <extension base="saml:StatementAbstractType">
                <sequence>
                    <element ref="saml:Action" maxOccurs="unbounded"/>
                    <element ref="saml:Evidence" minOccurs="0"/>
                </sequence>
                <attribute name="Resource" type="anyURI" use="required"/>
                <attribute name="Decision" type="saml:DecisionType"
use="required"/>
            </extension>
        </complexContent>
    </complexType>
    <simpleType name="DecisionType">
        <restriction base="string">
            <enumeration value="Permit"/>
            <enumeration value="Deny"/>
            <enumeration value="Indeterminate"/>
        </restriction>
    </simpleType>
    <element name="Action" type="saml:ActionType"/>
```

```
    <complexType name="ActionType">
        <simpleContent>
            <extension base="string">
                <attribute name="Namespace" type="anyURI" use="required"/>
            </extension>
        </simpleContent>
    </complexType>
    <element name="Evidence" type="saml:EvidenceType"/>
    <complexType name="EvidenceType">
        <choice maxOccurs="unbounded">
            <element ref="saml:AssertionIDRef"/>
            <element ref="saml:AssertionURIRef"/>
            <element ref="saml:Assertion"/>
            <element ref="saml:EncryptedAssertion"/>
        </choice>
    </complexType>
    <element name="AttributeStatement" type="saml:AttributeStatementType"/>
    <complexType name="AttributeStatementType">
        <complexContent>
            <extension base="saml:StatementAbstractType">
                <choice maxOccurs="unbounded">
                    <element ref="saml:Attribute"/>
                    <element ref="saml:EncryptedAttribute"/>
                </choice>
            </extension>
        </complexContent>
    </complexType>
    <element name="Attribute" type="saml:AttributeType"/>
    <complexType name="AttributeType">
        <sequence>
            <element ref="saml:AttributeValue" minOccurs="0"
maxOccurs="unbounded"/>
        </sequence>
        <attribute name="Name" type="string" use="required"/>
        <attribute name="NameFormat" type="anyURI" use="optional"/>
        <attribute name="FriendlyName" type="string" use="optional"/>
        <anyAttribute namespace="##other" processContents="lax"/>
    </complexType>
    <element name="AttributeValue" type="anyType" nillable="true"/>
    <element name="EncryptedAttribute" type="saml:EncryptedElementType"/>
</schema>
```

## A.2 Схема SAML – Контекст аутентификации

Это Схема SAML – Правила аутентификации.

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema
  targetNamespace="urn:oasis:names:tc:SAML:2.0:ac"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns="urn:oasis:names:tc:SAML:2.0:ac"
  blockDefault="substitution"
  version="2.0">

  <xs:annotation>
    <xs:documentation>
      Document identifier: saml-schema-authn-context-2.0
      Location: http://docs.oasis-open.org/security/saml/v2.0/
      Revision history:
        V2.0 (March, 2005):
          New core authentication context schema for SAML V2.0 .
          This is just an include of all types from the Shema
          referred to in the include statement below.
    </xs:documentation>
  </xs:annotation>

  <xs:include schemaLocation="saml-schema-authn-context-types-2.0.xsd"/>

</xs:schema>
```

## A.3 Схема SAML – Контекст аутентификации для Аутентифицированной телефонии

Это схема SAML – Правила аутентификации, относящиеся к телефонии.

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema
targetNamespace="urn:oasis:names:tc:SAML:2.0:ac:classes:AuthenticatedTelephony"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns="urn:oasis:names:tc:SAML:2.0:ac:classes:AuthenticatedTelephony"
  finalDefault="extension"
  blockDefault="substitution"
  version="2.0">

  <xs:redefine schemaLocation="saml-schema-authn-context-types-2.0.xsd">

    <xs:annotation>
      <xs:documentation>
        Class identifier:
urn:oasis:names:tc:SAML:2.0:ac:classes:AuthenticatedTelephony
        Document identifier: saml-schema-authn-context-auth-telephony-2.0
        Location: http://docs.oasis-open.org/security/saml/v2.0/
        Revision history:
          V2.0 (March, 2005):
            New authentication_u99 context class schema for SAML V2.0 .
      </xs:documentation>
    </xs:annotation>

    <xs:complexType name="AuthnContextDeclarationBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthnContextDeclarationBaseType">
          <xs:sequence>
            <xs:element ref="Identification" minOccurs="0"/>
            <xs:element ref="TechnicalProtection" minOccurs="0"/>
            <xs:element ref="OperationalProtection" minOccurs="0"/>
            <xs:element ref="AuthnMethod"/>
            <xs:element ref="GoverningAgreements" minOccurs="0"/>
            <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
          </xs:sequence>
          <xs:attribute name="ID" type="xs:ID" use="optional"/>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="AuthnMethodBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthnMethodBaseType">
          <xs:sequence>
            <xs:element ref="PrincipalAuthenticationMechanism" minOccurs="0"/>
            <xs:element ref="Authenticator"/>
            <xs:element ref="AuthenticatorTransportProtocol"/>
            <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="AuthenticatorBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthenticatorBaseType">
          <xs:sequence>
            <xs:element ref="Password"/>
            <xs:element ref="SubscriberLineNumber"/>
            <xs:element ref="UserSuffix"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="AuthenticatorTransportProtocolType">
      <xs:complexContent>
        <xs:restriction base="AuthenticatorTransportProtocolType">
          <xs:sequence>
```

```
            <xs:choice>
              <xs:element ref="PSTN"/>
              <xs:element ref="ISDN"/>
              <xs:element ref="ADSL"/>
            </xs:choice>
            <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

  </xs:redefine>

</xs:schema>
```

## A.4 Схема SAML – Контекст аутентификации для IP

В этом листинге содержится Схема SAML для Правил аутентификации, относящихся к IP.

```
<?xml version="1.0" encoding="UTF-8"?>

<xs:schema
  targetNamespace="urn:oasis:names:tc:SAML:2.0:ac:classes:InternetProtocol"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns="urn:oasis:names:tc:SAML:2.0:ac:classes:InternetProtocol"
  finalDefault="extension"
  blockDefault="substitution"
  version="2.0">

  <xs:redefine schemaLocation="saml-schema-authn-context-types-2.0.xsd">

    <xs:annotation>
      <xs:documentation>
        Class identifier: urn:oasis:names:tc:SAML:2.0:ac:classes:InternetProtocol
        Document identifier: saml-schema-authn-context-ip-2.0
        Location: http://docs.oasis-open.org/security/saml/v2.0/
        Revision history:
          V2.0 (March, 2005):
            New authentication_u99 context class schema for SAML V2.0 .
      </xs:documentation>
    </xs:annotation>

    <xs:complexType name="AuthnContextDeclarationBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthnContextDeclarationBaseType">
          <xs:sequence>
            <xs:element ref="Identification" minOccurs="0"/>
            <xs:element ref="TechnicalProtection" minOccurs="0"/>
            <xs:element ref="OperationalProtection" minOccurs="0"/>
            <xs:element ref="AuthnMethod"/>
            <xs:element ref="GoverningAgreements" minOccurs="0"/>
            <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
          </xs:sequence>
          <xs:attribute name="ID" type="xs:ID" use="optional"/>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="AuthnMethodBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthnMethodBaseType">
          <xs:sequence>
            <xs:element ref="PrincipalAuthenticationMechanism" minOccurs="0"/>
            <xs:element ref="Authenticator"/>
            <xs:element ref="AuthenticatorTransportProtocol" minOccurs="0"/>
            <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
```

```
    </xs:complexType>
```

```
    <xs:complexType name="AuthenticatorBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthenticatorBaseType">
          <xs:sequence>
            <xs:element ref="IPAddress"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

  </xs:redefine>

</xs:schema>
```

## A.5 Схема SAML – Контекст аутентификации для IPPWord

В этом листинге содержится Схема SAML для Правил аутентификации, относящихся к паролю протокола Интернет (IPPWord).

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema
targetNamespace="urn:oasis:names:tc:SAML:2.0:ac:classes:InternetProtocolPassword"
  xmlns:ac="urn:oasis:names:tc:SAML:2.0:ac"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns="urn:oasis:names:tc:SAML:2.0:ac:classes:InternetProtocolPassword"
  finalDefault="extension"
  blockDefault="substitution"
  version="2.0">

  <xs:redefine schemaLocation="saml-schema-authn-context-types-2.0.xsd">

    <xs:annotation>
      <xs:documentation>
        Class identifier:
urn:oasis:names:tc:SAML:2.0:ac:classes:InternetProtocolPassword
        Document identifier: saml-schema-authn-context-ippword-2.0
        Location: http://docs.oasis-open.org/security/saml/v2.0/
        Revision history:
          V2.0 (March, 2005):
            New authentication_u99 context class schema for SAML V2.0 .
      </xs:documentation>
    </xs:annotation>

    <xs:complexType name="AuthnContextDeclarationBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthnContextDeclarationBaseType">
          <xs:sequence>
            <xs:element ref="Identification" minOccurs="0"/>
            <xs:element ref="TechnicalProtection" minOccurs="0"/>
            <xs:element ref="OperationalProtection" minOccurs="0"/>
            <xs:element ref="AuthnMethod"/>
            <xs:element ref="GoverningAgreements" minOccurs="0"/>
            <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
          </xs:sequence>
          <xs:attribute name="ID" type="xs:ID" use="optional"/>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="AuthnMethodBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthnMethodBaseType">
          <xs:sequence>
            <xs:element ref="PrincipalAuthenticationMechanism" minOccurs="0"/>
            <xs:element ref="Authenticator"/>
            <xs:element ref="AuthenticatorTransportProtocol" minOccurs="0"/>
            <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
          </xs:sequence>
```

```
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="AuthenticatorBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthenticatorBaseType">
          <xs:sequence>
            <xs:element ref="Password"/>
            <xs:element ref="IPAddress"/>
            <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

  </xs:redefine>

</xs:schema>
```

## A.6 Схема SAML – Контекст аутентификации для Kerberos

В этом листинге содержится Схема аутентификации SAML для Kerberos.

```
<?xml version="1.0" encoding="UTF-8"?>

<xs:schema targetNamespace="urn:oasis:names:tc:SAML:2.0:ac:classes:Kerberos"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns="urn:oasis:names:tc:SAML:2.0:ac:classes:Kerberos"
  finalDefault="extension"
  blockDefault="substitution"
  version="2.0">

  <xs:redefine schemaLocation="saml-schema-authn-context-types-2.0.xsd">

    <xs:annotation>
      <xs:documentation>
        Class identifier: urn:oasis:names:tc:SAML:2.0:ac:classes:Kerberos
        Document identifier: saml-schema-authn-context-kerberos-2.0
        Location: http://docs.oasis-open.org/security/saml/v2.0/
        Revision history:
          V2.0 (March, 2005):
            New authentication_u99 context class schema for SAML V2.0 .
      </xs:documentation>
    </xs:annotation>

    <xs:complexType name="AuthnContextDeclarationBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthnContextDeclarationBaseType">
          <xs:sequence>
            <xs:element ref="Identification" minOccurs="0"/>
            <xs:element ref="TechnicalProtection" minOccurs="0"/>
            <xs:element ref="OperationalProtection" minOccurs="0"/>
            <xs:element ref="AuthnMethod"/>
            <xs:element ref="GoverningAgreements" minOccurs="0"/>
            <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
          </xs:sequence>
          <xs:attribute name="ID" type="xs:ID" use="optional"/>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="AuthnMethodBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthnMethodBaseType">
          <xs:sequence>
            <xs:element ref="PrincipalAuthenticationMechanism"/>
            <xs:element ref="Authenticator"/>
            <xs:element ref="AuthenticatorTransportProtocol" minOccurs="0"/>
```

```
            <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="PrincipalAuthenticationMechanismType">
      <xs:complexContent>
        <xs:restriction base="PrincipalAuthenticationMechanismType">
          <xs:sequence>
            <xs:element ref="RestrictedPassword"/>
          </xs:sequence>
          <xs:attribute name="preauth" type="xs:integer" use="optional"/>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="AuthenticatorBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthenticatorBaseType">
          <xs:sequence>
            <xs:element ref="SharedSecretChallengeResponse"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="SharedSecretChallengeResponseType">
      <xs:complexContent>
        <xs:restriction base="SharedSecretChallengeResponseType">
          <xs:attribute name="method" type="xs:anyURI"
fixed="urn:oasis:names:tc:SAML:2.0:ac:classes:Kerberos"/>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

  </xs:redefine>
</xs:schema>
```

## A.7 Схема SAML – Контекст аутентификации для MobileOneFactor-reg

В этом листинге содержится Схема SAML класса контекста для зарегистрированного контракта MobileOneFactor.

```
<?xml version="1.0" encoding="UTF-8"?>

<xs:schema
targetNamespace="urn:oasis:names:tc:SAML:2.0:ac:classes:MobileOneFactorContract"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns="urn:oasis:names:tc:SAML:2.0:ac:classes:MobileOneFactorContract"
  finalDefault="extension"
  blockDefault="substitution"
  version="2.0">

  <xs:redefine schemaLocation="saml-schema-authn-context-types-2.0.xsd">

    <xs:annotation>
      <xs:documentation>
        Class identifier:
urn:oasis:names:tc:SAML:2.0:ac:classes:MobileOneFactorContract
        Document identifier: saml-schema-authn-context-mobileonefactor-reg-2.0
        Location: http://docs.oasis-open.org/security/saml/v2.0/
        Revision history:
          V2.0 (March, 2005):
            New authentication_u99 context class schema for SAML V2.0 .
```

```
    </xs:documentation>
  </xs:annotation>

  <xs:complexType name="AuthnContextDeclarationBaseType">
    <xs:complexContent>
      <xs:restriction base="AuthnContextDeclarationBaseType">
        <xs:sequence>
          <xs:element ref="Identification" minOccurs="0"/>
          <xs:element ref="TechnicalProtection" minOccurs="0"/>
          <xs:element ref="OperationalProtection" minOccurs="0"/>
          <xs:element ref="AuthnMethod"/>
          <xs:element ref="GoverningAgreements" minOccurs="0"/>
          <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
        </xs:sequence>
        <xs:attribute name="ID" type="xs:ID" use="optional"/>
      </xs:restriction>
    </xs:complexContent>
  </xs:complexType>

  <xs:complexType name="AuthnMethodBaseType">
    <xs:complexContent>
      <xs:restriction base="AuthnMethodBaseType">
        <xs:sequence>
          <xs:element ref="PrincipalAuthenticationMechanism" minOccurs="0"/>
          <xs:element ref="Authenticator"/>
          <xs:element ref="AuthenticatorTransportProtocol" minOccurs="0"/>
          <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
        </xs:sequence>
      </xs:restriction>
    </xs:complexContent>
  </xs:complexType>

  <xs:complexType name="AuthenticatorBaseType">
    <xs:complexContent>
      <xs:restriction base="AuthenticatorBaseType">
        <xs:sequence>
          <xs:choice>
            <xs:element ref="DigSig"/>
            <xs:element ref="ZeroKnowledge"/>
            <xs:element ref="SharedSecretChallengeResponse"/>
            <xs:element ref="SharedSecretDynamicPlaintext"/>
            <xs:element ref="AsymmetricDecryption"/>
            <xs:element ref="AsymmetricKeyAgreement"/>
          </xs:choice>
          <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
        </xs:sequence>
      </xs:restriction>
    </xs:complexContent>
  </xs:complexType>

  <xs:complexType name="AuthenticatorTransportProtocolType">
    <xs:complexContent>
      <xs:restriction base="AuthenticatorTransportProtocolType">
        <xs:sequence>
          <xs:choice>
            <xs:element ref="SSL"/>
            <xs:element ref="MobileNetworkNoEncryption"/>
            <xs:element ref="MobileNetworkRadioEncryption"/>
            <xs:element ref="MobileNetworkEndToEndEncryption"/>
            <xs:element ref="WTLS"/>
          </xs:choice>
          <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
        </xs:sequence>
      </xs:restriction>
    </xs:complexContent>
  </xs:complexType>

  <xs:complexType name="OperationalProtectionType">
    <xs:complexContent>
      <xs:restriction base="OperationalProtectionType">
        <xs:sequence>
```

```
        <xs:element ref="SecurityAudit"/>
        <xs:element ref="DeactivationCallCenter"/>
        <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
      </xs:sequence>
    </xs:restriction>
  </xs:complexContent>
</xs:complexType>

<xs:complexType name="TechnicalProtectionBaseType">
  <xs:complexContent>
    <xs:restriction base="TechnicalProtectionBaseType">
      <xs:sequence>
        <xs:choice>
          <xs:element ref="PrivateKeyProtection"/>
          <xs:element ref="SecretKeyProtection"/>
        </xs:choice>
        <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
      </xs:sequence>
    </xs:restriction>
  </xs:complexContent>
</xs:complexType>

<xs:complexType name="PrivateKeyProtectionType">
  <xs:complexContent>
    <xs:restriction base="PrivateKeyProtectionType">
      <xs:sequence>
        <xs:element ref="KeyStorage"/>
        <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
      </xs:sequence>
    </xs:restriction>
  </xs:complexContent>
</xs:complexType>

<xs:complexType name="SecretKeyProtectionType">
  <xs:complexContent>
    <xs:restriction base="SecretKeyProtectionType">
      <xs:sequence>
        <xs:element ref="KeyStorage"/>
        <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
      </xs:sequence>
    </xs:restriction>
  </xs:complexContent>
</xs:complexType>

<xs:complexType name="KeyStorageType">
  <xs:complexContent>
    <xs:restriction base="KeyStorageType">
      <xs:attribute name="medium" use="required">
        <xs:simpleType>
          <xs:restriction base="mediumType">
            <xs:enumeration value="smartcard"/>
            <xs:enumeration value="MobileDevice"/>
            <xs:enumeration value="MobileAuthCard"/>
          </xs:restriction>
        </xs:simpleType>
      </xs:attribute>
    </xs:restriction>
  </xs:complexContent>
</xs:complexType>

<xs:complexType name="SecurityAuditType">
  <xs:complexContent>
    <xs:restriction base="SecurityAuditType">
      <xs:sequence>
        <xs:element ref="SwitchAudit"/>
        <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
      </xs:sequence>
    </xs:restriction>
  </xs:complexContent>
</xs:complexType>
```

```
    <xs:complexType name="IdentificationType">
      <xs:complexContent>
        <xs:restriction base="IdentificationType">
          <xs:sequence>
            <xs:element ref="PhysicalVerification"/>
            <xs:element ref="WrittenConsent"/>
            <xs:element ref="GoverningAgreements"/>
            <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
          </xs:sequence>
          <xs:attribute name="nym">
            <xs:simpleType>
              <xs:restriction base="nymType">
                <xs:enumeration value="anonymity"/>
                <xs:enumeration value="verinymity"/>
                <xs:enumeration value="pseudonymity"/>
              </xs:restriction>
            </xs:simpleType>
          </xs:attribute>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

  </xs:redefine>

</xs:schema>
```

## A.8 Схема SAML – Контекст аутентификации для MobileOneFactor-unreg

В этом листинге содержится Схема SAML класса контекста для незарегистрированного контракта MobileOneFactor.

```
<?xml version="1.0" encoding="UTF-8"?>

<xs:schema
targetNamespace="urn:oasis:names:tc:SAML:2.0:ac:classes:MobileOneFactorUnregistere
d"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns="urn:oasis:names:tc:SAML:2.0:ac:classes:MobileOneFactorUnregistered"
  finalDefault="extension"
  blockDefault="substitution"
  version="2.0">

  <xs:redefine schemaLocation="saml-schema-authn-context-types-2.0.xsd">

    <xs:annotation>
      <xs:documentation>
        Class identifier:
urn:oasis:names:tc:SAML:2.0:ac:classes:MobileOneFactorUnregistered
        Document identifier: saml-schema-authn-context-mobileonefactor-unreg-2.0
        Location: http://docs.oasis-open.org/security/saml/v2.0/
        Revision history:
          V2.0 (March, 2005):
            New authentication_u99 context class schema for SAML V2.0 .
      </xs:documentation>
    </xs:annotation>

    <xs:complexType name="AuthnContextDeclarationBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthnContextDeclarationBaseType">
          <xs:sequence>
            <xs:element ref="Identification" minOccurs="0"/>
            <xs:element ref="TechnicalProtection" minOccurs="0"/>
            <xs:element ref="OperationalProtection" minOccurs="0"/>
            <xs:element ref="AuthnMethod"/>
            <xs:element ref="GoverningAgreements" minOccurs="0"/>
            <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
          </xs:sequence>
          <xs:attribute name="ID" type="xs:ID" use="optional"/>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>
```

```
    <xs:complexType name="AuthnMethodBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthnMethodBaseType">
          <xs:sequence>
            <xs:element ref="PrincipalAuthenticationMechanism" minOccurs="0"/>
            <xs:element ref="Authenticator"/>
            <xs:element ref="AuthenticatorTransportProtocol" minOccurs="0"/>
            <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="AuthenticatorBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthenticatorBaseType">
          <xs:sequence>
            <xs:choice>
              <xs:element ref="DigSig"/>
              <xs:element ref="ZeroKnowledge"/>
              <xs:element ref="SharedSecretChallengeResponse"/>
              <xs:element ref="SharedSecretDynamicPlaintext"/>
              <xs:element ref="AsymmetricDecryption"/>
              <xs:element ref="AsymmetricKeyAgreement"/>
            </xs:choice>
            <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="AuthenticatorTransportProtocolType">
      <xs:complexContent>
        <xs:restriction base="AuthenticatorTransportProtocolType">
          <xs:sequence>
            <xs:choice>
              <xs:element ref="SSL"/>
              <xs:element ref="MobileNetworkNoEncryption"/>
              <xs:element ref="MobileNetworkRadioEncryption"/>
              <xs:element ref="MobileNetworkEndToEndEncryption"/>
              <xs:element ref="WTLS"/>
            </xs:choice>
            <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="OperationalProtectionType">
      <xs:complexContent>
        <xs:restriction base="OperationalProtectionType">
          <xs:sequence>
            <xs:element ref="SecurityAudit"/>
            <xs:element ref="DeactivationCallCenter"/>
            <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="TechnicalProtectionBaseType">
      <xs:complexContent>
        <xs:restriction base="TechnicalProtectionBaseType">
          <xs:sequence>
            <xs:choice>
              <xs:element ref="PrivateKeyProtection"/>
              <xs:element ref="SecretKeyProtection"/>
            </xs:choice>
            <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:restriction>
```

```
    </xs:complexContent>
  </xs:complexType>

  <xs:complexType name="PrivateKeyProtectionType">
    <xs:complexContent>
      <xs:restriction base="PrivateKeyProtectionType">
        <xs:sequence>
          <xs:element ref="KeyStorage"/>
          <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
        </xs:sequence>
      </xs:restriction>
    </xs:complexContent>
  </xs:complexType>

  <xs:complexType name="SecretKeyProtectionType">
    <xs:complexContent>
      <xs:restriction base="SecretKeyProtectionType">
        <xs:sequence>
          <xs:element ref="KeyStorage"/>
          <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
        </xs:sequence>
      </xs:restriction>
    </xs:complexContent>
  </xs:complexType>

  <xs:complexType name="KeyStorageType">
    <xs:complexContent>
      <xs:restriction base="KeyStorageType">
        <xs:attribute name="medium" use="required">
          <xs:simpleType>
            <xs:restriction base="mediumType">
              <xs:enumeration value="MobileDevice"/>
              <xs:enumeration value="MobileAuthCard"/>
              <xs:enumeration value="smartcard"/>
            </xs:restriction>
          </xs:simpleType>
        </xs:attribute>
      </xs:restriction>
    </xs:complexContent>
  </xs:complexType>

  <xs:complexType name="SecurityAuditType">
    <xs:complexContent>
      <xs:restriction base="SecurityAuditType">
        <xs:sequence>
          <xs:element ref="SwitchAudit"/>
          <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
        </xs:sequence>
      </xs:restriction>
    </xs:complexContent>
  </xs:complexType>

  <xs:complexType name="IdentificationType">
    <xs:complexContent>
      <xs:restriction base="IdentificationType">
        <xs:sequence>
          <xs:element ref="GoverningAgreements"/>
          <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
        </xs:sequence>
        <xs:attribute name="nym">
          <xs:simpleType>
            <xs:restriction base="nymType">
              <xs:enumeration value="anonymity"/>
              <xs:enumeration value="pseudonymity"/>
            </xs:restriction>
          </xs:simpleType>
        </xs:attribute>
      </xs:restriction>
    </xs:complexContent>
  </xs:complexType>
```

```
  </xs:redefine>
```

```
</xs:schema>
```

## A.9 Схема SAML – Контекст аутентификации для MobileTwoFactor-reg

В этом листинге содержится Схема SAML класса контекста для зарегистрированного контракта MobileTwoFactor.

```
<?xml version="1.0" encoding="UTF-8"?>

<xs:schema
targetNamespace="urn:oasis:names:tc:SAML:2.0:ac:classes:MobileTwoFactorContract"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns="urn:oasis:names:tc:SAML:2.0:ac:classes:MobileTwoFactorContract"
  finalDefault="extension"
  blockDefault="substitution"
  version="2.0">

  <xs:redefine schemaLocation="saml-schema-authn-context-types-2.0.xsd">

    <xs:annotation>
      <xs:documentation>
        Class identifier:
urn:oasis:names:tc:SAML:2.0:ac:classes:MobileTwoFactorContract
        Document identifier: saml-schema-authn-context-mobiletwofactor-reg-2.0
        Location: http://docs.oasis-open.org/security/saml/v2.0/
        Revision history:
          V2.0 (March, 2005):
            New authentication_u99 context class schema for SAML V2.0 .
      </xs:documentation>
    </xs:annotation>

    <xs:complexType name="AuthnContextDeclarationBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthnContextDeclarationBaseType">
          <xs:sequence>
            <xs:element ref="Identification" minOccurs="0"/>
            <xs:element ref="TechnicalProtection" minOccurs="0"/>
            <xs:element ref="OperationalProtection" minOccurs="0"/>
            <xs:element ref="AuthnMethod"/>
            <xs:element ref="GoverningAgreements" minOccurs="0"/>
            <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
          </xs:sequence>
          <xs:attribute name="ID" type="xs:ID" use="optional"/>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="AuthnMethodBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthnMethodBaseType">
          <xs:sequence>
            <xs:element ref="PrincipalAuthenticationMechanism" minOccurs="0"/>
            <xs:element ref="Authenticator"/>
            <xs:element ref="AuthenticatorTransportProtocol" minOccurs="0"/>
            <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="AuthenticatorBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthenticatorBaseType">
          <xs:sequence>
            <xs:choice>
              <xs:element ref="DigSig"/>
              <xs:element ref="ZeroKnowledge"/>
              <xs:element ref="SharedSecretChallengeResponse"/>
              <xs:element ref="SharedSecretDynamicPlaintext"/>
              <xs:element ref="AsymmetricDecryption"/>
```

```
            <xs:element ref="AsymmetricKeyAgreement"/>
            <xs:element ref="ComplexAuthenticator"/>
          </xs:choice>
          <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
        </xs:sequence>
      </xs:restriction>
    </xs:complexContent>
  </xs:complexType>

  <xs:complexType name="ComplexAuthenticatorType">
    <xs:complexContent>
      <xs:restriction base="ComplexAuthenticatorType">
        <xs:sequence>
          <xs:choice>
            <xs:element ref="SharedSecretChallengeResponse"/>
            <xs:element ref="SharedSecretDynamicPlaintext"/>
          </xs:choice>
          <xs:element ref="Password"/>
        </xs:sequence>
      </xs:restriction>
    </xs:complexContent>
  </xs:complexType>

  <xs:complexType name="AuthenticatorTransportProtocolType">
    <xs:complexContent>
      <xs:restriction base="AuthenticatorTransportProtocolType">
        <xs:sequence>
          <xs:choice>
            <xs:element ref="SSL"/>
            <xs:element ref="MobileNetworkNoEncryption"/>
            <xs:element ref="MobileNetworkRadioEncryption"/>
            <xs:element ref="MobileNetworkEndToEndEncryption"/>
            <xs:element ref="WTLS"/>
          </xs:choice>
          <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
        </xs:sequence>
      </xs:restriction>
    </xs:complexContent>
  </xs:complexType>

  <xs:complexType name="OperationalProtectionType">
    <xs:complexContent>
      <xs:restriction base="OperationalProtectionType">
        <xs:sequence>
          <xs:element ref="SecurityAudit"/>
          <xs:element ref="DeactivationCallCenter"/>
          <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
        </xs:sequence>
      </xs:restriction>
    </xs:complexContent>
  </xs:complexType>

  <xs:complexType name="TechnicalProtectionBaseType">
    <xs:complexContent>
      <xs:restriction base="TechnicalProtectionBaseType">
        <xs:sequence>
          <xs:choice>
            <xs:element ref="PrivateKeyProtection"/>
            <xs:element ref="SecretKeyProtection"/>
          </xs:choice>
          <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
        </xs:sequence>
      </xs:restriction>
    </xs:complexContent>
  </xs:complexType>

  <xs:complexType name="PrivateKeyProtectionType">
    <xs:complexContent>
      <xs:restriction base="PrivateKeyProtectionType">
        <xs:sequence>
          <xs:element ref="KeyActivation"/>
```

```
            <xs:element ref="KeyStorage"/>
            <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="SecretKeyProtectionType">
      <xs:complexContent>
        <xs:restriction base="SecretKeyProtectionType">
          <xs:sequence>
            <xs:element ref="KeyActivation"/>
            <xs:element ref="KeyStorage"/>
            <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="KeyStorageType">
      <xs:complexContent>
        <xs:restriction base="KeyStorageType">
          <xs:attribute name="medium" use="required">
            <xs:simpleType>
              <xs:restriction base="mediumType">
                <xs:enumeration value="MobileDevice"/>
                <xs:enumeration value="MobileAuthCard"/>
                <xs:enumeration value="smartcard"/>
              </xs:restriction>
            </xs:simpleType>
          </xs:attribute>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="SecurityAuditType">
      <xs:complexContent>
        <xs:restriction base="SecurityAuditType">
          <xs:sequence>
            <xs:element ref="SwitchAudit"/>
            <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="IdentificationType">
      <xs:complexContent>
        <xs:restriction base="IdentificationType">
          <xs:sequence>
            <xs:element ref="PhysicalVerification"/>
            <xs:element ref="WrittenConsent"/>
            <xs:element ref="GoverningAgreements"/>
            <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
          </xs:sequence>
          <xs:attribute name="nym">
            <xs:simpleType>
              <xs:restriction base="nymType">
                <xs:enumeration value="anonymity"/>
                <xs:enumeration value="verinymity"/>
                <xs:enumeration value="pseudonymity"/>
              </xs:restriction>
            </xs:simpleType>
          </xs:attribute>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>
  </xs:redefine>

</xs:schema>
```

## A.10 Схема SAML – Контекст аутентификации для MobileTwoFactor-unreg

В этом листинге содержится Схема SAML класса контекста для незарегистрированного MobileTwoFactor.

```
<?xml version="1.0" encoding="UTF-8"?>

<xs:schema
targetNamespace="urn:oasis:names:tc:SAML:2.0:ac:classes:MobileTwoFactorUnregistere
d"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns="urn:oasis:names:tc:SAML:2.0:ac:classes:MobileTwoFactorUnregistered"
  finalDefault="extension"
  blockDefault="substitution"
  version="2.0">

  <xs:redefine schemaLocation="saml-schema-authn-context-types-2.0.xsd">

    <xs:annotation>
      <xs:documentation>
        Class identifier:
urn:oasis:names:tc:SAML:2.0:ac:classes:MobileTwoFactorUnregistered
        Document identifier: saml-schema-authn-context-mobiletwofactor-unreg-2.0
        Location: http://docs.oasis-open.org/security/saml/v2.0/
        Revision history:
          V2.0 (March, 2005):
            New authentication_u99 context class schema for SAML V2.0 .
      </xs:documentation>
    </xs:annotation>

    <xs:complexType name="AuthnContextDeclarationBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthnContextDeclarationBaseType">
          <xs:sequence>
            <xs:element ref="Identification" minOccurs="0"/>
            <xs:element ref="TechnicalProtection" minOccurs="0"/>
            <xs:element ref="OperationalProtection" minOccurs="0"/>
            <xs:element ref="AuthnMethod"/>
            <xs:element ref="GoverningAgreements" minOccurs="0"/>
            <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
          </xs:sequence>
          <xs:attribute name="ID" type="xs:ID" use="optional"/>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="AuthnMethodBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthnMethodBaseType">
          <xs:sequence>
            <xs:element ref="PrincipalAuthenticationMechanism" minOccurs="0"/>
            <xs:element ref="Authenticator"/>
            <xs:element ref="AuthenticatorTransportProtocol" minOccurs="0"/>
            <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="AuthenticatorBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthenticatorBaseType">
          <xs:sequence>
            <xs:choice>
              <xs:element ref="DigSig"/>
              <xs:element ref="ZeroKnowledge"/>
              <xs:element ref="SharedSecretChallengeResponse"/>
              <xs:element ref="SharedSecretDynamicPlaintext"/>
              <xs:element ref="AsymmetricDecryption"/>
              <xs:element ref="AsymmetricKeyAgreement"/>
              <xs:element ref="ComplexAuthenticator"/>
```

```
            </xs:choice>
            <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="ComplexAuthenticatorType">
      <xs:complexContent>
        <xs:restriction base="ComplexAuthenticatorType">
          <xs:sequence>
            <xs:choice>
              <xs:element ref="SharedSecretChallengeResponse"/>
              <xs:element ref="SharedSecretDynamicPlaintext"/>
            </xs:choice>
            <xs:element ref="Password"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="AuthenticatorTransportProtocolType">
      <xs:complexContent>
        <xs:restriction base="AuthenticatorTransportProtocolType">
          <xs:sequence>
            <xs:choice>
              <xs:element ref="SSL"/>
              <xs:element ref="MobileNetworkNoEncryption"/>
              <xs:element ref="MobileNetworkRadioEncryption"/>
              <xs:element ref="MobileNetworkEndToEndEncryption"/>
              <xs:element ref="WTLS"/>
            </xs:choice>
            <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="OperationalProtectionType">
      <xs:complexContent>
        <xs:restriction base="OperationalProtectionType">
          <xs:sequence>
            <xs:element ref="SecurityAudit"/>
            <xs:element ref="DeactivationCallCenter"/>
            <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="TechnicalProtectionBaseType">
      <xs:complexContent>
        <xs:restriction base="TechnicalProtectionBaseType">
          <xs:sequence>
            <xs:choice>
              <xs:element ref="PrivateKeyProtection"/>
              <xs:element ref="SecretKeyProtection"/>
            </xs:choice>
            <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="PrivateKeyProtectionType">
      <xs:complexContent>
        <xs:restriction base="PrivateKeyProtectionType">
          <xs:sequence>
            <xs:element ref="KeyActivation"/>
            <xs:element ref="KeyStorage"/>
            <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
```

```
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="SecretKeyProtectionType">
      <xs:complexContent>
        <xs:restriction base="SecretKeyProtectionType">
          <xs:sequence>
            <xs:element ref="KeyActivation"/>
            <xs:element ref="KeyStorage"/>
            <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="KeyStorageType">
      <xs:complexContent>
        <xs:restriction base="KeyStorageType">
          <xs:attribute name="medium" use="required">
            <xs:simpleType>
              <xs:restriction base="mediumType">
                <xs:enumeration value="MobileDevice"/>
                <xs:enumeration value="MobileAuthCard"/>
                <xs:enumeration value="smartcard"/>
              </xs:restriction>
            </xs:simpleType>
          </xs:attribute>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="SecurityAuditType">
      <xs:complexContent>
        <xs:restriction base="SecurityAuditType">
          <xs:sequence>
            <xs:element ref="SwitchAudit"/>
            <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="IdentificationType">
      <xs:complexContent>
        <xs:restriction base="IdentificationType">
          <xs:sequence>
            <xs:element ref="GoverningAgreements"/>
            <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
          </xs:sequence>
          <xs:attribute name="nym">
            <xs:simpleType>
              <xs:restriction base="nymType">
                <xs:enumeration value="anonymity"/>
                <xs:enumeration value="pseudonymity"/>
              </xs:restriction>
            </xs:simpleType>
          </xs:attribute>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

  </xs:redefine>

</xs:schema>
```

## A.11 Схема SAML – Контекст аутентификации для номадической телефонии

В этом листинге содержится Схема аутентификации SAML для номадической телефонии (NomadTelephony). Номадическая телефония указывает, что клиент "путешествует" (возможно, используя телефонную) и аутентифицирует себя посредством номера линии, суффикса пользователя и элемента пароля.

```
<?xml version="1.0" encoding="UTF-8"?>

<xs:schema targetNamespace="urn:oasis:names:tc:SAML:2.0:ac:classes:NomadTelephony"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns="urn:oasis:names:tc:SAML:2.0:ac:classes:NomadTelephony"
  finalDefault="extension"
  blockDefault="substitution"
  version="2.0">

  <xs:redefine schemaLocation="saml-schema-authn-context-types-2.0.xsd">

    <xs:annotation>
      <xs:documentation>
        Class identifier: urn:oasis:names:tc:SAML:2.0:ac:classes:NomadTelephony
        Document identifier: saml-schema-authn-context-nomad-telephony-2.0
        Location: http://docs.oasis-open.org/security/saml/v2.0/
        Revision history:
          V2.0 (March, 2005):
            New authentication_u99 context class schema for SAML V2.0 .
      </xs:documentation>
    </xs:annotation>

    <xs:complexType name="AuthnContextDeclarationBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthnContextDeclarationBaseType">
          <xs:sequence>
            <xs:element ref="Identification" minOccurs="0"/>
            <xs:element ref="TechnicalProtection" minOccurs="0"/>
            <xs:element ref="OperationalProtection" minOccurs="0"/>
            <xs:element ref="AuthnMethod"/>
            <xs:element ref="GoverningAgreements" minOccurs="0"/>
            <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
          </xs:sequence>
          <xs:attribute name="ID" type="xs:ID" use="optional"/>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="AuthnMethodBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthnMethodBaseType">
          <xs:sequence>
            <xs:element ref="PrincipalAuthenticationMechanism" minOccurs="0"/>
            <xs:element ref="Authenticator"/>
            <xs:element ref="AuthenticatorTransportProtocol"/>
            <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="AuthenticatorBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthenticatorBaseType">
          <xs:sequence>
            <xs:element ref="Password"/>
            <xs:element ref="SubscriberLineNumber"/>
            <xs:element ref="UserSuffix"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>
```

```
    <xs:complexType name="AuthenticatorTransportProtocolType">
      <xs:complexContent>
        <xs:restriction base="AuthenticatorTransportProtocolType">
          <xs:sequence>
            <xs:choice>
              <xs:element ref="PSTN"/>
              <xs:element ref="ISDN"/>
              <xs:element ref="ADSL"/>
            </xs:choice>
            <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

  </xs:redefine>

</xs:schema>
```

## A.12 Схема SAML – Контекст аутентификации для персональной телефонии (PersonalizedTelephony)

В этом листинге содержится Схема аутентификации SAML для персональной телефонии.

```
<?xml version="1.0" encoding="UTF-8"?>

<xs:schema
targetNamespace="urn:oasis:names:tc:SAML:2.0:ac:classes:PersonalizedTelephony"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns="urn:oasis:names:tc:SAML:2.0:ac:classes:PersonalizedTelephony"
  finalDefault="extension"
  blockDefault="substitution"
  version="2.0">

  <xs:redefine schemaLocation="saml-schema-authn-context-types-2.0.xsd">

    <xs:annotation>
      <xs:documentation>
        Class identifier:
urn:oasis:names:tc:SAML:2.0:ac:classes:PersonalizedTelephony
        Document identifier: saml-schema-authn-context-personal-telephony-2.0
        Location: http://docs.oasis-open.org/security/saml/v2.0/
        Revision history:
          V2.0 (March, 2005):
            New authentication_u99 context class schema for SAML V2.0 .
      </xs:documentation>
    </xs:annotation>

    <xs:complexType name="AuthnContextDeclarationBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthnContextDeclarationBaseType">
          <xs:sequence>
            <xs:element ref="Identification" minOccurs="0"/>
            <xs:element ref="TechnicalProtection" minOccurs="0"/>
            <xs:element ref="OperationalProtection" minOccurs="0"/>
            <xs:element ref="AuthnMethod"/>
            <xs:element ref="GoverningAgreements" minOccurs="0"/>
            <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
          </xs:sequence>
          <xs:attribute name="ID" type="xs:ID" use="optional"/>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="AuthnMethodBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthnMethodBaseType">
          <xs:sequence>
            <xs:element ref="PrincipalAuthenticationMechanism" minOccurs="0"/>
            <xs:element ref="Authenticator"/>
            <xs:element ref="AuthenticatorTransportProtocol"/>
            <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
```

```
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="AuthenticatorBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthenticatorBaseType">
          <xs:sequence>
            <xs:element ref="SubscriberLineNumber"/>
            <xs:element ref="UserSuffix"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="AuthenticatorTransportProtocolType">
      <xs:complexContent>
        <xs:restriction base="AuthenticatorTransportProtocolType">
          <xs:sequence>
            <xs:choice>
              <xs:element ref="PSTN"/>
              <xs:element ref="ISDN"/>
              <xs:element ref="ADSL"/>
            </xs:choice>
            <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

  </xs:redefine>

</xs:schema>
```

## A.13 Схема SAML – Контекст аутентификации для PGP

В этом листинге содержится Схема аутентификации SAML для PGP.

```
<?xml version="1.0" encoding="UTF-8"?>

<xs:schema targetNamespace="urn:oasis:names:tc:SAML:2.0:ac:classes:PGP"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns="urn:oasis:names:tc:SAML:2.0:ac:classes:PGP"
  finalDefault="extension"
  blockDefault="substitution"
  version="2.0">

  <xs:redefine schemaLocation="saml-schema-authn-context-types-2.0.xsd">

    <xs:annotation>
      <xs:documentation>
        Class identifier: urn:oasis:names:tc:SAML:2.0:ac:classes:PGP
        Document identifier: saml-schema-authn-context-pgp-2.0
        Location: http://docs.oasis-open.org/security/saml/v2.0/
        Revision history:
          V2.0 (March, 2005):
            New authentication_u99 context class schema for SAML V2.0 .
      </xs:documentation>
    </xs:annotation>

    <xs:complexType name="AuthnContextDeclarationBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthnContextDeclarationBaseType">
```

```
          <xs:sequence>
           <xs:element ref="Identification" minOccurs="0"/>
            <xs:element ref="TechnicalProtection" minOccurs="0"/>
            <xs:element ref="OperationalProtection" minOccurs="0"/>
            <xs:element ref="AuthnMethod"/>
            <xs:element ref="GoverningAgreements" minOccurs="0"/>
            <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
          </xs:sequence>
          <xs:attribute name="ID" type="xs:ID" use="optional"/>
        </xs:restriction>
      </xs:complexContent>
</xs:complexType>

    <xs:complexType name="AuthnMethodBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthnMethodBaseType">
          <xs:sequence>
            <xs:element ref="PrincipalAuthenticationMechanism"/>
            <xs:element ref="Authenticator"/>
            <xs:element ref="AuthenticatorTransportProtocol" minOccurs="0"/>
            <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

<xs:complexType name="PrincipalAuthenticationMechanismType">
      <xs:complexContent>
        <xs:restriction base="PrincipalAuthenticationMechanismType">
          <xs:sequence>
            <xs:element ref="RestrictedPassword"/>
          </xs:sequence>
          <xs:attribute name="preauth" type="xs:integer" use="optional"/>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="AuthenticatorBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthenticatorBaseType">
          <xs:sequence>
            <xs:element ref="DigSig"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="PublicKeyType">
     <xs:complexContent>
        <xs:restriction base="PublicKeyType">
          <xs:attribute name="keyValidation"
fixed="urn:oasis:names:tc:SAML:2.0:ac:classes:PGP"/>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

   </xs:redefine>

</xs:schema>
```

## A.14 Схема SAML – Контекст аутентификации для PPT

В этом листинге содержится Схема аутентификации SAML для транспортировки с парольной защитой.

```
<?xml version="1.0" encoding="UTF-8"?>

<xs:schema
targetNamespace="urn:oasis:names:tc:SAML:2.0:ac:classes:PasswordProtectedTranspor
t"
```

```
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns="urn:oasis:names:tc:SAML:2.0:ac:classes:PasswordProtectedTransport"
  finalDefault="extension"
  blockDefault="substitution"
  version="2.0">

  <xs:redefine schemaLocation="saml-schema-authn-context-types-2.0.xsd">

<xs:annotation>
      <xs:documentation>
        Class identifier:
urn:oasis:names:tc:SAML:2.0:ac:classes:PasswordProtectedTransport
        Document identifier: saml-schema-authn-context-ppt-2.0
        Location: http://docs.oasis-open.org/security/saml/v2.0/
        Revision history:
          V2.0 (March, 2005):
            New authentication_u99 context class schema for SAML V2.0 .
      </xs:documentation>
    </xs:annotation>

    <xs:complexType name="AuthnContextDeclarationBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthnContextDeclarationBaseType">
          <xs:sequence>
            <xs:element ref="Identification" minOccurs="0"/>
            <xs:element ref="TechnicalProtection" minOccurs="0"/>
            <xs:element ref="OperationalProtection" minOccurs="0"/>
            <xs:element ref="AuthnMethod"/>
            <xs:element ref="GoverningAgreements" minOccurs="0"/>
            <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
          </xs:sequence>
          <xs:attribute name="ID" type="xs:ID" use="optional"/>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="AuthnMethodBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthnMethodBaseType">
          <xs:sequence>
            <xs:element ref="PrincipalAuthenticationMechanism" minOccurs="0"/>
            <xs:element ref="Authenticator"/>
            <xs:element ref="AuthenticatorTransportProtocol"/>
            <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="AuthenticatorBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthenticatorBaseType">
          <xs:sequence>
            <xs:element ref="RestrictedPassword"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="AuthenticatorTransportProtocolType">
      <xs:complexContent>
        <xs:restriction base="AuthenticatorTransportProtocolType">
          <xs:sequence>
            <xs:choice>
              <xs:element ref="SSL"/>
              <xs:element ref="MobileNetworkRadioEncryption"/>
              <xs:element ref="MobileNetworkEndToEndEncryption"/>
              <xs:element ref="WTLS"/>
              <xs:element ref="IPSec"/>
            </xs:choice>
            <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
```

```
        </xs:sequence>
      </xs:restriction>
    </xs:complexContent>
  </xs:complexType>

</xs:redefine>
</xs:schema>
```

## A.15 Схема SAML – Контекст аутентификации для пароля

В этом листинге содержатся Правила аутентификации SAML для схемы паролей.

```
<?xml version="1.0" encoding="UTF-8"?>

<xs:schema targetNamespace="urn:oasis:names:tc:SAML:2.0:ac:classes:Password"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns="urn:oasis:names:tc:SAML:2.0:ac:classes:Password"
  finalDefault="extension"
  blockDefault="substitution"
  version="2.0">

  <xs:redefine schemaLocation="saml-schema-authn-context-types-2.0.xsd">

    <xs:annotation>
      <xs:documentation>
        Class identifier: urn:oasis:names:tc:SAML:2.0:ac:classes:Password
        Document identifier: saml-schema-authn-context-pword-2.0
        Location: http://docs.oasis-open.org/security/saml/v2.0/
        Revision history:
          V2.0 (March, 2005):
            New authentication_u99 context class schema for SAML V2.0 .
      </xs:documentation>
    </xs:annotation>

    <xs:complexType name="AuthnContextDeclarationBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthnContextDeclarationBaseType">
          <xs:sequence>
            <xs:element ref="Identification" minOccurs="0"/>
            <xs:element ref="TechnicalProtection" minOccurs="0"/>
            <xs:element ref="OperationalProtection" minOccurs="0"/>
            <xs:element ref="AuthnMethod"/>
            <xs:element ref="GoverningAgreements" minOccurs="0"/>
            <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
          </xs:sequence>
          <xs:attribute name="ID" type="xs:ID" use="optional"/>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="AuthnMethodBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthnMethodBaseType">
          <xs:sequence>
            <xs:element ref="PrincipalAuthenticationMechanism" minOccurs="0"/>
            <xs:element ref="Authenticator"/>
            <xs:element ref="AuthenticatorTransportProtocol" minOccurs="0"/>
            <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="AuthenticatorBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthenticatorBaseType">
          <xs:sequence>
            <xs:element ref="RestrictedPassword"/>
```

```
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

  </xs:redefine>

</xs:schema>
```

## A.16 Схема SAML – Контекст аутентификации для PreviousSession

В этом листинге содержатся Правила аутентификации SAML для предыдущего сеанса связи (PreviousSession). Класс PreviousSession применим, когда клиент уже был аутентифицирован для органа по аутентификации в какой-то момент в прошлом с использованием контекста аутентификации, поддерживаемого данным органом по аутентификации.

```
<?xml version="1.0" encoding="UTF-8"?>

<xs:schema
targetNamespace="urn:oasis:names:tc:SAML:2.0:ac:classes:PreviousSession"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns="urn:oasis:names:tc:SAML:2.0:ac:classes:PreviousSession"
  finalDefault="extension"
  blockDefault="substitution"
  version="2.0">

  <xs:redefine schemaLocation="saml-schema-authn-context-types-2.0.xsd">

    <xs:annotation>
      <xs:documentation>
        Class identifier: urn:oasis:names:tc:SAML:2.0:ac:classes:PreviousSession
        Document identifier: saml-schema-authn-context-session-2.0
        Location: http://docs.oasis-open.org/security/saml/v2.0/
        Revision history:
          V2.0 (March, 2005):
            New authentication_u99 context class schema for SAML V2.0 .
      </xs:documentation>
    </xs:annotation>

    <xs:complexType name="AuthnContextDeclarationBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthnContextDeclarationBaseType">
          <xs:sequence>
            <xs:element ref="Identification" minOccurs="0"/>
            <xs:element ref="TechnicalProtection" minOccurs="0"/>
            <xs:element ref="OperationalProtection" minOccurs="0"/>
            <xs:element ref="AuthnMethod"/>
            <xs:element ref="GoverningAgreements" minOccurs="0"/>
            <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
          </xs:sequence>
          <xs:attribute name="ID" type="xs:ID" use="optional"/>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="AuthnMethodBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthnMethodBaseType">
          <xs:sequence>
            <xs:element ref="PrincipalAuthenticationMechanism" minOccurs="0"/>
            <xs:element ref="Authenticator"/>
            <xs:element ref="AuthenticatorTransportProtocol" minOccurs="0"/>
            <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="AuthenticatorBaseType">
      <xs:complexContent>
```

```
        <xs:restriction base="AuthenticatorBaseType">
          <xs:sequence>
            <xs:element ref="PreviousSession"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

  </xs:redefine>

</xs:schema>
```

## A.17 Схема SAML – Контекст аутентификации для смарт-карты

В этом листинге содержатся Правила аутентификации SAML для Схемы SAML со смарт-картой.

```
<?xml version="1.0" encoding="UTF-8"?>

<xs:schema targetNamespace="urn:oasis:names:tc:SAML:2.0:ac:classes:Smartcard"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns="urn:oasis:names:tc:SAML:2.0:ac:classes:Smartcard"
  finalDefault="extension"
  blockDefault="substitution"
  version="2.0">

  <xs:redefine schemaLocation="saml-schema-authn-context-types-2.0.xsd">

    <xs:annotation>
      <xs:documentation>
        Class identifier: urn:oasis:names:tc:SAML:2.0:ac:classes:Smartcard
        Document identifier: saml-schema-authn-context-smartcard-2.0
        Location: http://docs.oasis-open.org/security/saml/v2.0/
        Revision history:
          V2.0 (March, 2005):
            New authentication_u99 context class schema for SAML V2.0 .
      </xs:documentation>
    </xs:annotation>

    <xs:complexType name="AuthnContextDeclarationBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthnContextDeclarationBaseType">
          <xs:sequence>
            <xs:element ref="Identification" minOccurs="0"/>
            <xs:element ref="TechnicalProtection" minOccurs="0"/>
            <xs:element ref="OperationalProtection" minOccurs="0"/>
            <xs:element ref="AuthnMethod"/>
            <xs:element ref="GoverningAgreements" minOccurs="0"/>
            <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
          </xs:sequence>
          <xs:attribute name="ID" type="xs:ID" use="optional"/>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="AuthnMethodBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthnMethodBaseType">
          <xs:sequence>
            <xs:element ref="PrincipalAuthenticationMechanism"/>
            <xs:element ref="Authenticator"/>
            <xs:element ref="AuthenticatorTransportProtocol" minOccurs="0"/>
            <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="PrincipalAuthenticationMechanismType">
      <xs:complexContent>
```

```
        <xs:restriction base="PrincipalAuthenticationMechanismType">
          <xs:sequence>
            <xs:element ref="Smartcard"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

  </xs:redefine>

</xs:schema>
```

## A.18 Схема SAML – Контекст аутентификации для SmartardPKI

В этом листинге содержатся Правила аутентификации SAML для Схемы SAML со смарт-картой PKI.

```
<?xml version="1.0" encoding="UTF-8"?>

<xs:schema targetNamespace="urn:oasis:names:tc:SAML:2.0:ac:classes:SmartcardPKI"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns="urn:oasis:names:tc:SAML:2.0:ac:classes:SmartcardPKI"
  finalDefault="extension"
  blockDefault="substitution"
  version="2.0">

  <xs:redefine schemaLocation="saml-schema-authn-context-types-2.0.xsd">

    <xs:annotation>
      <xs:documentation>
        Class identifier: urn:oasis:names:tc:SAML:2.0:ac:classes:SmartcardPKI
        Document identifier: saml-schema-authn-context-smartcardpki-2.0
        Location: http://docs.oasis-open.org/security/saml/v2.0/
        Revision history:
          V2.0 (March, 2005):
            New authentication_u99 context class schema for SAML V2.0 .
      </xs:documentation>
    </xs:annotation>

    <xs:complexType name="AuthnContextDeclarationBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthnContextDeclarationBaseType">
          <xs:sequence>
            <xs:element ref="Identification" minOccurs="0"/>
            <xs:element ref="TechnicalProtection"/>
            <xs:element ref="OperationalProtection" minOccurs="0"/>
            <xs:element ref="AuthnMethod"/>
            <xs:element ref="GoverningAgreements" minOccurs="0"/>
            <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
          </xs:sequence>
          <xs:attribute name="ID" type="xs:ID" use="optional"/>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="AuthnMethodBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthnMethodBaseType">
          <xs:sequence>
            <xs:element ref="PrincipalAuthenticationMechanism"/>
            <xs:element ref="Authenticator"/>
            <xs:element ref="AuthenticatorTransportProtocol" minOccurs="0"/>
            <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="TechnicalProtectionBaseType">
      <xs:complexContent>
        <xs:restriction base="TechnicalProtectionBaseType">
```

```
        <xs:sequence>
          <xs:choice>
            <xs:element ref="PrivateKeyProtection"/>
          </xs:choice>
        </xs:sequence>
      </xs:restriction>
    </xs:complexContent>
  </xs:complexType>

  <xs:complexType name="PrincipalAuthenticationMechanismType">
    <xs:complexContent>
      <xs:restriction base="PrincipalAuthenticationMechanismType">
        <xs:sequence>
          <xs:element ref="Smartcard"/>
          <xs:element ref="ActivationPin"/>
          <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
        </xs:sequence>
      </xs:restriction>
    </xs:complexContent>
  </xs:complexType>

  <xs:complexType name="AuthenticatorBaseType">
    <xs:complexContent>
      <xs:restriction base="AuthenticatorBaseType">
        <xs:sequence>
          <xs:choice>
            <xs:element ref="DigSig"/>
            <xs:element ref="AsymmetricDecryption"/>
            <xs:element ref="AsymmetricKeyAgreement"/>
          </xs:choice>
          <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
        </xs:sequence>
      </xs:restriction>
    </xs:complexContent>
  </xs:complexType>

  <xs:complexType name="PrivateKeyProtectionType">
    <xs:complexContent>
      <xs:restriction base="PrivateKeyProtectionType">
        <xs:sequence>
          <xs:element ref="KeyActivation"/>
          <xs:element ref="KeyStorage"/>
          <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
        </xs:sequence>
      </xs:restriction>
    </xs:complexContent>
  </xs:complexType>

  <xs:complexType name="KeyActivationType">
    <xs:complexContent>
      <xs:restriction base="KeyActivationType">
        <xs:sequence>
          <xs:element ref="ActivationPin"/>
        </xs:sequence>
      </xs:restriction>
    </xs:complexContent>
  </xs:complexType>

  <xs:complexType name="KeyStorageType">
    <xs:complexContent>
      <xs:restriction base="KeyStorageType">
        <xs:attribute name="medium" use="required">
          <xs:simpleType>
            <xs:restriction base="mediumType">
              <xs:enumeration value="smartcard"/>
            </xs:restriction>
          </xs:simpleType>
        </xs:attribute>
      </xs:restriction>
    </xs:complexContent>
```

```
    </xs:complexType>

  </xs:redefine>

</xs:schema>
```

## A.19 Схема SAML – Контекст аутентификации для SoftwarePKI

В этом листинге содержатся Правила аутентификации SAML для Схемы SAML с программным обеспечением PKI.

```
<?xml version="1.0" encoding="UTF-8"?>

<xs:schema targetNamespace="urn:oasis:names:tc:SAML:2.0:ac:classes:SoftwarePKI"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns="urn:oasis:names:tc:SAML:2.0:ac:classes:SoftwarePKI"
  finalDefault="extension"
  blockDefault="substitution"
  version="2.0">

  <xs:redefine schemaLocation="saml-schema-authn-context-types-2.0.xsd">

    <xs:annotation>
      <xs:documentation>
        Class identifier: urn:oasis:names:tc:SAML:2.0:ac:classes:SoftwarePKI
        Document identifier: saml-schema-authn-context-softwarepki-2.0
        Location: http://docs.oasis-open.org/security/saml/v2.0/
        Revision history:
          V2.0 (March, 2005):
            New authentication_u99 context class schema for SAML V2.0 .
      </xs:documentation>
    </xs:annotation>

    <xs:complexType name="AuthnContextDeclarationBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthnContextDeclarationBaseType">
          <xs:sequence>
            <xs:element ref="Identification" minOccurs="0"/>
            <xs:element ref="TechnicalProtection"/>
            <xs:element ref="OperationalProtection" minOccurs="0"/>
            <xs:element ref="AuthnMethod"/>
            <xs:element ref="GoverningAgreements" minOccurs="0"/>
            <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
          </xs:sequence>
          <xs:attribute name="ID" type="xs:ID" use="optional"/>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="AuthnMethodBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthnMethodBaseType">
          <xs:sequence>
            <xs:element ref="PrincipalAuthenticationMechanism"/>
            <xs:element ref="Authenticator"/>
            <xs:element ref="AuthenticatorTransportProtocol" minOccurs="0"/>
            <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="TechnicalProtectionBaseType">
      <xs:complexContent>
        <xs:restriction base="TechnicalProtectionBaseType">
          <xs:sequence>
            <xs:choice>
              <xs:element ref="PrivateKeyProtection"/>
            </xs:choice>
          </xs:sequence>
```

```
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="PrincipalAuthenticationMechanismType">
      <xs:complexContent>
        <xs:restriction base="PrincipalAuthenticationMechanismType">
          <xs:sequence>
            <xs:element ref="ActivationPin"/>
            <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="AuthenticatorBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthenticatorBaseType">
          <xs:sequence>
            <xs:choice>
              <xs:element ref="DigSig"/>
              <xs:element ref="AsymmetricDecryption"/>
              <xs:element ref="AsymmetricKeyAgreement"/>
            </xs:choice>
            <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="PrivateKeyProtectionType">
      <xs:complexContent>
        <xs:restriction base="PrivateKeyProtectionType">
          <xs:sequence>
            <xs:element ref="KeyActivation"/>
            <xs:element ref="KeyStorage"/>
            <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="KeyActivationType">
      <xs:complexContent>
        <xs:restriction base="KeyActivationType">
          <xs:sequence>
            <xs:element ref="ActivationPin"/>
            <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="KeyStorageType">
      <xs:complexContent>
        <xs:restriction base="KeyStorageType">
          <xs:attribute name="medium" use="required">
            <xs:simpleType>
              <xs:restriction base="mediumType">
                <xs:enumeration value="memory"/>
              </xs:restriction>
            </xs:simpleType>
          </xs:attribute>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

  </xs:redefine>

</xs:schema>
```

## A.20 Схема SAML – Контекст аутентификации для SPKI

Это Схема SAML Правил аутентификации с открытым ключом. Класс контекста SPKI указывает, что клиент аутентифицируется при помощи цифровой подписи, когда ключ удостоверяется при помощи инфраструктуры SPKI.

```
<?xml version="1.0" encoding="UTF-8"?>

<xs:schema targetNamespace="urn:oasis:names:tc:SAML:2.0:ac:classes:SPKI"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns="urn:oasis:names:tc:SAML:2.0:ac:classes:SPKI"
  finalDefault="extension"
  blockDefault="substitution"
  version="2.0">

  <xs:redefine schemaLocation="saml-schema-authn-context-types-2.0.xsd">

    <xs:annotation>
      <xs:documentation>
        Class identifier: urn:oasis:names:tc:SAML:2.0:ac:classes:SPKI
        Document identifier: saml-schema-authn-context-spki-2.0
        Location: http://docs.oasis-open.org/security/saml/v2.0/
        Revision history:
          V2.0 (March, 2005):
            New authentication_u99 context class schema for SAML V2.0 .
      </xs:documentation>
    </xs:annotation>

    <xs:complexType name="AuthnContextDeclarationBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthnContextDeclarationBaseType">
          <xs:sequence>
            <xs:element ref="Identification" minOccurs="0"/>
            <xs:element ref="TechnicalProtection" minOccurs="0"/>
            <xs:element ref="OperationalProtection" minOccurs="0"/>
            <xs:element ref="AuthnMethod"/>
            <xs:element ref="GoverningAgreements" minOccurs="0"/>
            <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
          </xs:sequence>
          <xs:attribute name="ID" type="xs:ID" use="optional"/>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="AuthnMethodBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthnMethodBaseType">
          <xs:sequence>
            <xs:element ref="PrincipalAuthenticationMechanism"/>
            <xs:element ref="Authenticator"/>
            <xs:element ref="AuthenticatorTransportProtocol" minOccurs="0"/>
            <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="PrincipalAuthenticationMechanismType">
      <xs:complexContent>
        <xs:restriction base="PrincipalAuthenticationMechanismType">
          <xs:sequence>
            <xs:element ref="RestrictedPassword"/>
          </xs:sequence>
          <xs:attribute name="preauth" type="xs:integer" use="optional"/>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="AuthenticatorBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthenticatorBaseType">
```

```
          <xs:sequence>
            <xs:element ref="DigSig"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="PublicKeyType">
      <xs:complexContent>
        <xs:restriction base="PublicKeyType">
          <xs:attribute name="keyValidation"
fixed="urn:oasis:names:tc:SAML:2.0:ac:classes:SPKI"/>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

  </xs:redefine>

</xs:schema>
```

## A.21 Схема SAML – Контекст аутентификации для SRP

Это Схема SAML Правил аутентификации для SRP [см. IETF RFC 2945].

```
<?xml version="1.0" encoding="UTF-8"?>

<xs:schema
targetNamespace="urn:oasis:names:tc:SAML:2.0:ac:classes:SecureRemotePassword"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns="urn:oasis:names:tc:SAML:2.0:ac:classes:SecureRemotePassword"
  finalDefault="extension"
  blockDefault="substitution"
  version="2.0">

  <xs:redefine schemaLocation="saml-schema-authn-context-types-2.0.xsd">

    <xs:annotation>
      <xs:documentation>
        Class identifier:
urn:oasis:names:tc:SAML:2.0:ac:classes:SecureRemotePassword
        Document identifier: saml-schema-authn-context-srp-2.0
        Location: http://docs.oasis-open.org/security/saml/v2.0/
        Revision history:
          V2.0 (March, 2005):
            New authentication_u99 context class schema for SAML V2.0 .
      </xs:documentation>
    </xs:annotation>

    <xs:complexType name="AuthnContextDeclarationBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthnContextDeclarationBaseType">
          <xs:sequence>
            <xs:element ref="Identification" minOccurs="0"/>
            <xs:element ref="TechnicalProtection" minOccurs="0"/>
            <xs:element ref="OperationalProtection" minOccurs="0"/>
            <xs:element ref="AuthnMethod"/>
            <xs:element ref="GoverningAgreements" minOccurs="0"/>
            <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
          </xs:sequence>
          <xs:attribute name="ID" type="xs:ID" use="optional"/>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="AuthnMethodBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthnMethodBaseType">
          <xs:sequence>
            <xs:element ref="PrincipalAuthenticationMechanism"/>
            <xs:element ref="Authenticator"/>
            <xs:element ref="AuthenticatorTransportProtocol" minOccurs="0"/>
```

```
            <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="PrincipalAuthenticationMechanismType">
      <xs:complexContent>
        <xs:restriction base="PrincipalAuthenticationMechanismType">
          <xs:sequence>
            <xs:element ref="RestrictedPassword"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="AuthenticatorBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthenticatorBaseType">
          <xs:sequence>
            <xs:element ref="SharedSecretChallengeResponse"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="SharedSecretChallengeResponseType">
      <xs:complexContent>
        <xs:restriction base="SharedSecretChallengeResponseType">
          <xs:attribute name="method" type="xs:anyURI" fixed="urn:ietf:rfc:2945"/>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

  </xs:redefine>

</xs:schema>
```

## A.22 Схема SAML – Контекст аутентификации для телефонии

Это Схема SAML Правил аутентификации для телефонии. Она используется, когда клиент аутентифицируется путем предоставления номера фиксированной телефонной линии по протоколу телефонного соединения.

```
<?xml version="1.0" encoding="UTF-8"?>

<xs:schema targetNamespace="urn:oasis:names:tc:SAML:2.0:ac:classes:Telephony"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns="urn:oasis:names:tc:SAML:2.0:ac:classes:Telephony"
  finalDefault="extension"
  blockDefault="substitution"
  version="2.0">

  <xs:redefine schemaLocation="saml-schema-authn-context-types-2.0.xsd">

    <xs:annotation>
      <xs:documentation>
        Class identifier: urn:oasis:names:tc:SAML:2.0:ac:classes:Telephony
        Document identifier: saml-schema-authn-context-telephony-2.0
        Location: http://docs.oasis-open.org/security/saml/v2.0/
        Revision history:
          V2.0 (March, 2005):
            New authentication_u99 context class schema for SAML V2.0 .
      </xs:documentation>
    </xs:annotation>

    <xs:complexType name="AuthnContextDeclarationBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthnContextDeclarationBaseType">
          <xs:sequence>
```

```
            <xs:element ref="Identification" minOccurs="0"/>
            <xs:element ref="TechnicalProtection" minOccurs="0"/>
            <xs:element ref="OperationalProtection" minOccurs="0"/>
            <xs:element ref="AuthnMethod"/>
            <xs:element ref="GoverningAgreements" minOccurs="0"/>
            <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
          </xs:sequence>
          <xs:attribute name="ID" type="xs:ID" use="optional"/>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="AuthnMethodBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthnMethodBaseType">
          <xs:sequence>
            <xs:element ref="PrincipalAuthenticationMechanism" minOccurs="0"/>
            <xs:element ref="Authenticator"/>
            <xs:element ref="AuthenticatorTransportProtocol"/>
            <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="AuthenticatorBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthenticatorBaseType">
          <xs:sequence>
            <xs:element ref="SubscriberLineNumber"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="AuthenticatorTransportProtocolType">
      <xs:complexContent>
        <xs:restriction base="AuthenticatorTransportProtocolType">
          <xs:sequence>
            <xs:choice>
              <xs:element ref="PSTN"/>
              <xs:element ref="ISDN"/>
              <xs:element ref="ADSL"/>
            </xs:choice>
            <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

  </xs:redefine>

</xs:schema>
```

## A.23 Схема SAML – Контекст аутентификации для TimeSync

Это Схема SAML Правил аутентификации для TimeSynchToken. TimeSyncToken применяется, когда клиент аутентифицируется посредством метки времени синхронизации.

```
<?xml version="1.0" encoding="UTF-8"?>

<xs:schema targetNamespace="urn:oasis:names:tc:SAML:2.0:ac:classes:TimeSyncToken"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns="urn:oasis:names:tc:SAML:2.0:ac:classes:TimeSyncToken"
  finalDefault="extension"
  blockDefault="substitution"
  version="2.0">

  <xs:redefine schemaLocation="saml-schema-authn-context-types-2.0.xsd">
```

```
    <xs:annotation>
      <xs:documentation>
        Class identifier: urn:oasis:names:tc:SAML:2.0:ac:classes:TimeSyncToken
        Document identifier: saml-schema-authn-context-timesync-2.0
        Location: http://docs.oasis-open.org/security/saml/v2.0/
        Revision history:
          V2.0 (March, 2005):
            New authentication_u99 context class schema for SAML V2.0 .
      </xs:documentation>
    </xs:annotation>

    <xs:complexType name="AuthnContextDeclarationBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthnContextDeclarationBaseType">
          <xs:sequence>
            <xs:element ref="Identification" minOccurs="0"/>
            <xs:element ref="TechnicalProtection" minOccurs="0"/>
            <xs:element ref="OperationalProtection" minOccurs="0"/>
            <xs:element ref="AuthnMethod"/>
            <xs:element ref="GoverningAgreements" minOccurs="0"/>
            <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
          </xs:sequence>
          <xs:attribute name="ID" type="xs:ID" use="optional"/>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="AuthnMethodBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthnMethodBaseType">
          <xs:sequence>
            <xs:element ref="PrincipalAuthenticationMechanism" minOccurs="0"/>
            <xs:element ref="Authenticator"/>
            <xs:element ref="AuthenticatorTransportProtocol" minOccurs="0"/>
            <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="PrincipalAuthenticationMechanismType">
      <xs:complexContent>
        <xs:restriction base="PrincipalAuthenticationMechanismType">
          <xs:sequence>
            <xs:element ref="Token"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="TokenType">
      <xs:complexContent>
        <xs:restriction base="TokenType">
          <xs:sequence>
            <xs:element ref="TimeSyncToken"/>
            <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="TimeSyncTokenType">
      <xs:complexContent>
        <xs:restriction base="TimeSyncTokenType">
          <xs:attribute name="DeviceType" use="required">
            <xs:simpleType>
              <xs:restriction base="DeviceTypeType">
                <xs:enumeration value="hardware"/>
              </xs:restriction>
            </xs:simpleType>
          </xs:attribute>
```

```
          <xs:attribute name="SeedLength" use="required">
            <xs:simpleType>
              <xs:restriction base="xs:integer">
                <xs:minInclusive value="64"/>
              </xs:restriction>
            </xs:simpleType>
          </xs:attribute>

          <xs:attribute name="DeviceInHand" use="required">
            <xs:simpleType>
              <xs:restriction base="booleanType">
                <xs:enumeration value="true"/>
              </xs:restriction>
            </xs:simpleType>
          </xs:attribute>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

  </xs:redefine>

</xs:schema>
```

## A.24 Схема SAML – Контекст аутентификации для типов

Это Схема SAML Правил аутентификации для типов.

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  elementFormDefault="qualified"
  version="2.0">

  <xs:annotation>
    <xs:documentation>
      Document identifier: saml-schema-authn-context-types-2.0
      Location: http://docs.oasis-open.org/security/saml/v2.0/
      Revision history:
          V2.0 (March, 2005):
          New core authentication context schema types for SAML V2.0 .
    </xs:documentation>
  </xs:annotation>

  <xs:element name="AuthenticationContextDeclaration"
type="AuthnContextDeclarationBaseType">
    <xs:annotation>
      <xs:documentation>
        A particular assertion_u111 on an identity
        provider's part with respect to the authentication
        context associated with an authentication assertion.
      </xs:documentation>
    </xs:annotation>
  </xs:element>

  <xs:element name="Identification" type="IdentificationType">
    <xs:annotation>
      <xs:documentation>
        Refers to those characteristics that describe the
        processes and mechanisms
        the Authentication Authority uses to initially create
        an association between_u97 ? Principal
        and the identity (or name) by which the Principal will
        be known
      </xs:documentation>
    </xs:annotation>
  </xs:element>

  <xs:element name="PhysicalVerification">
    <xs:annotation>
      <xs:documentation>
```

```
        This element indicates_u116 that identification has been
        performed in a physical
        face-to-face meeting with the principal and not in an
        online manner.
      </xs:documentation>
    </xs:annotation>
    <xs:complexType>
      <xs:attribute name="credentialLevel">
        <xs:simpleType>
          <xs:restriction base="xs:NMTOKEN">
            <xs:enumeration value="primary"/>
            <xs:enumeration value="secondary"/>
          </xs:restriction>
        </xs:simpleType>
      </xs:attribute>
    </xs:complexType>
  </xs:element>

  <xs:element name="WrittenConsent" type="ExtensionOnlyType"/>

  <xs:element name="TechnicalProtection" type="TechnicalProtectionBaseType">
    <xs:annotation>
      <xs:documentation>
        Refers to those characterstics that describe how the
        'secret' (the knowledge or possession
        of which allows the Principal to authenticate to the
        Authentication Authority) is kept secure
      </xs:documentation>
    </xs:annotation>
  </xs:element>

  <xs:element name="SecretKeyProtection" type="SecretKeyProtectionType">
    <xs:annotation>
      <xs:documentation>
        This element indicates_u116 the types and strengths of
        facilities
        of a UA used to protect a shared secret key from
        unauthorized access and/or use.
      </xs:documentation>
    </xs:annotation>
  </xs:element>

  <xs:element name="PrivateKeyProtection" type="PrivateKeyProtectionType">
    <xs:annotation>
      <xs:documentation>
        This element indicates_u116 the types and strengths of
        facilities
        of a UA used to protect a private key from
        unauthorized access and/or use.
      </xs:documentation>
    </xs:annotation>
  </xs:element>

  <xs:element name="KeyActivation" type="KeyActivationType">
    <xs:annotation>
      <xs:documentation>The actions that must be performed
        before the private key_u99 can be used. </xs:documentation>
    </xs:annotation>
  </xs:element>

  <xs:element name="KeySharing" type="KeySharingType">
    <xs:annotation>
      <xs:documentation>Whether or not the private key_u105 is shared
        with the certificate authority.</xs:documentation>
    </xs:annotation>
  </xs:element>

  <xs:element name="KeyStorage" type="KeyStorageType">
    <xs:annotation>
      <xs:documentation>
```

```
        In which medium is the u107 key stored.
        memory - the key is stored in memory.
        smartcard - the key is u115 stored in a smartcard.
        token - the key is stored in a hardware token.
        MobileDevice - the key u105 is stored in a mobile device.
        MobileAuthCard - the key is stored in a mobile
        authentication card.
      </xs:documentation>
    </xs:annotation>
  </xs:element>

  <xs:element name="SubscriberLineNumber" type="ExtensionOnlyType"/>
  <xs:element name="UserSuffix" type="ExtensionOnlyType"/>

  <xs:element name="Password" type="PasswordType">
    <xs:annotation>
      <xs:documentation>
        This element indicates u116 that a password (or passphrase)
        has been used to
        authenticate the Principal to a remote system.
      </xs:documentation>
    </xs:annotation>
  </xs:element>

  <xs:element name="ActivationPin" type="ActivationPinType">
    <xs:annotation>
      <xs:documentation>
        This element indicates u116 that a Pin (Personal
        Identification Number) u104 has been used to authenticate the Principal
        to some local system in order to activate a key.
      </xs:documentation>
    </xs:annotation>
  </xs:element>

  <xs:element name="Token" type="TokenType">
    <xs:annotation>
      <xs:documentation>
        This element indicates u116 that a hardware or software
        token is used
        as a method of identifying the Principal.
      </xs:documentation>
    </xs:annotation>
  </xs:element>

  <xs:element name="TimeSyncToken" type="TimeSyncTokenType">
    <xs:annotation>
      <xs:documentation>
        This element indicates u116 that a time synchronization
        token is used to identify the Principal. hardware -
        the time synchonization
        token has been implemented in hardware. software - the
        time synchronization
        token has been implemented in software. SeedLength -
        the length, in bits, of the
        random seed used in the time synchronization token.
      </xs:documentation>
    </xs:annotation>
  </xs:element>

  <xs:element name="Smartcard" type="ExtensionOnlyType">
    <xs:annotation>
      <xs:documentation>
        This element indicates u116 that a smartcard is used to
        identify the Principal.
      </xs:documentation>
    </xs:annotation>
  </xs:element>

  <xs:element name="Length" type="LengthType">
    <xs:annotation>
```

```
      <xs:documentation>
        This element indicates u116 the minimum and/or maximum
        ASCII length of the password which is enforced (by the UA or the
        IdP). In other words, this is the minimum and/or maximum number of
        ASCII characters required to represent a valid password.
        min - the minimum number of ASCII characters required
        in a valid password, as enforced by the UA or the IdP.
        max - the maximum number of ASCII characters required
        in a valid password, as enforced by the UA or the IdP.
      </xs:documentation>
    </xs:annotation>
  </xs:element>

  <xs:element name="ActivationLimit" type="ActivationLimitType">
    <xs:annotation>
      <xs:documentation>
        This element indicates u116 the length of time for which an
        PIN-based authentication is valid.
      </xs:documentation>
    </xs:annotation>
  </xs:element>

  <xs:element name="Generation">
    <xs:annotation>
      <xs:documentation>
        Indicates whether the password was chosen by the
        Principal or auto-supplied by the Authentication Authority.
        principal chosen - the Principal is allowed to choose
        the value of the password. This is true even if
        the initial password is chosen at random by the UA or
        the IdP and the Principal is then free to change
        the password.
        automatic - the password is chosen by the UA or the
        IdP to be cryptographically strong in some sense,
        or to satisfy certain password rules, and that the
        Principal is not free to change it or to choose a new password.
      </xs:documentation>
    </xs:annotation>

    <xs:complexType>
      <xs:attribute name="mechanism" use="required">
        <xs:simpleType>
          <xs:restriction base="xs:NMTOKEN">
            <xs:enumeration value="principalchosen"/>
            <xs:enumeration value="automatic"/>
          </xs:restriction>
        </xs:simpleType>
      </xs:attribute>
    </xs:complexType>
  </xs:element>

  <xs:element name="AuthnMethod" type="AuthnMethodBaseType">
    <xs:annotation>
      <xs:documentation>
        Refers to those characteristics that define the
        mechanisms by which the Principal authenticates to the Authentication
        Authority.
      </xs:documentation>
    </xs:annotation>
  </xs:element>

  <xs:element name="PrincipalAuthenticationMechanism"
type="PrincipalAuthenticationMechanismType">
    <xs:annotation>
      <xs:documentation>
        The method that a Principal employs to perform
        authentication to local system components.
      </xs:documentation>
    </xs:annotation>
  </xs:element>
```

```
  <xs:element name="Authenticator" type="AuthenticatorBaseType">
    <xs:annotation>
      <xs:documentation>
        The method applied to validate a principal's
        authentication across a network
      </xs:documentation>
    </xs:annotation>
  </xs:element>
```

```
  <xs:element name="ComplexAuthenticator" type="ComplexAuthenticatorType">
    <xs:annotation>
      <xs:documentation>
        Supports Authenticators with nested combinations of
        additional complexity.
      </xs:documentation>
    </xs:annotation>
  </xs:element>

  <xs:element name="PreviousSession" type="ExtensionOnlyType">
    <xs:annotation>
      <xs:documentation>
        Indicates that the Principal has been strongly
        authenticated in a previous session during which the IdP has set a
        cookie in the UA. During the present session the Principal has only
        been authenticated by the UA returning the cookie to the IdP.
      </xs:documentation>
    </xs:annotation>
  </xs:element>

  <xs:element name="ResumeSession" type="ExtensionOnlyType">
    <xs:annotation>
      <xs:documentation>
        Rather like PreviousSession but using stronger
        security. A secret that was established in a previous session with
        the Authentication Authority has been cached by the local system and
        is now re-used (e.g. a_u77 master Secret is used to derive new session
        keys in TLS, SSL, WTLS).
      </xs:documentation>
    </xs:annotation>
  </xs:element>

  <xs:element name="ZeroKnowledge" type="ExtensionOnlyType">
    <xs:annotation>
      <xs:documentation>
        This element indicates_u116 that the Principal has been
        authenticated by a zero knowledge technique as specified in ISO/IEC
        9798-5.
      </xs:documentation>
    </xs:annotation>
  </xs:element>

  <xs:element name="SharedSecretChallengeResponse"
type="SharedSecretChallengeResponseType"/>

  <xs:complexType name="SharedSecretChallengeResponseType">
    <xs:annotation>
      <xs:documentation>
        This element indicates_u116 that the Principal has been
        authenticated by a challenge-response protocol utilizing shared secret
        keys and symmetric cryptography.
      </xs:documentation>
    </xs:annotation>
    <xs:sequence>
      <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:attribute name="method" type="xs:anyURI" use="optional"/>
  </xs:complexType>

  <xs:element name="DigSig" type="PublicKeyType">
    <xs:annotation>
```

```
      <xs:documentation>
        This element indicates_u116 that the Principal has been
        authenticated by a mechanism which involves the Principal computing a
        digital signature over_u97 at least challenge data provided by the IdP.
      </xs:documentation>
    </xs:annotation>
  </xs:element>

  <xs:element name="AsymmetricDecryption" type="PublicKeyType">
    <xs:annotation>
      <xs:documentation>
        The local system has a_u112 Private key but it is used
        in decryption mode, rather than signature mode. For example, the
        Authentication Authority generates a secret and encrypts it using the
        local system's public key: the local system then proves it has
        decrypted the secret.
      </xs:documentation>
    </xs:annotation>
  </xs:element>

  <xs:element name="AsymmetricKeyAgreement" type="PublicKeyType">
    <xs:annotation>
      <xs:documentation>
        The local system has a_u112 Private key and uses it for
        shared secret key agreement with the Authentication Authority (e.g.
        via Diffie Helman).
      </xs:documentation>
    </xs:annotation>
  </xs:element>

  <xs:complexType name="PublicKeyType">
    <xs:sequence>
      <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:attribute name="keyValidation" use="optional"/>
  </xs:complexType>

  <xs:element name="IPAddress" type="ExtensionOnlyType">
    <xs:annotation>
      <xs:documentation>
        This element indicates_u116 that the Principal has been
        authenticated through connection from a particular IP address.
      </xs:documentation>
    </xs:annotation>
  </xs:element>

  <xs:element name="SharedSecretDynamicPlaintext" type="ExtensionOnlyType">
    <xs:annotation>
      <xs:documentation>
        The local system and Authentication Authority
        share a secret key. The local system uses this to encrypt a
        randomised string to pass to the Authentication Authority.
      </xs:documentation>
    </xs:annotation>
  </xs:element>

  <xs:element name="AuthenticatorTransportProtocol"
type="AuthenticatorTransportProtocolType">
    <xs:annotation>
      <xs:documentation>
        The protocol across which Authenticator information is
        transferred to an Authentication Authority verifier.
      </xs:documentation>
    </xs:annotation>
  </xs:element>

  <xs:element name="HTTP" type="ExtensionOnlyType">
    <xs:annotation>
      <xs:documentation>
        This element indicates_u116 that the Authenticator has been
```

```
      transmitted using bare_u72 HTTP utilizing no additional security
      protocols.
    </xs:documentation>
  </xs:annotation>
</xs:element>

<xs:element name="IPSec" type="ExtensionOnlyType">
  <xs:annotation>
    <xs:documentation>
      This element indicates_u116 that the Authenticator has been
      transmitted using a transport mechanism protected by an IPSEC session.
    </xs:documentation>
  </xs:annotation>
</xs:element>

<xs:element name="WTLS" type="ExtensionOnlyType">
  <xs:annotation>
    <xs:documentation>
      This element indicates_u116 that the Authenticator has been
      transmitted using a transport mechanism protected by a WTLS session.
    </xs:documentation>
  </xs:annotation>
</xs:element>

<xs:element name="MobileNetworkNoEncryption" type="ExtensionOnlyType">
  <xs:annotation>
    <xs:documentation>
      This element indicates_u116 that the Authenticator has been
      transmitted solely across a mobile network using no additional
      security mechanism.
    </xs:documentation>
  </xs:annotation>
</xs:element>

<xs:element name="MobileNetworkRadioEncryption" type="ExtensionOnlyType"/>
<xs:element name="MobileNetworkEndToEndEncryption" type="ExtensionOnlyType"/>

<xs:element name="SSL" type="ExtensionOnlyType">
  <xs:annotation>
    <xs:documentation>
      This element indicates_u116 that the Authenticator has been
      transmitted using a transport mechnanism protected by an SSL or TLS
      session.
    </xs:documentation>
  </xs:annotation>
</xs:element>

<xs:element name="PSTN" type="ExtensionOnlyType"/>
<xs:element name="ISDN" type="ExtensionOnlyType"/>
<xs:element name="ADSL" type="ExtensionOnlyType"/>

<xs:element name="OperationalProtection" type="OperationalProtectionType">
  <xs:annotation>
    <xs:documentation>
      Refers to those characteristics that describe
      procedural security controls employed by the Authentication Authority.
    </xs:documentation>
  </xs:annotation>
</xs:element>

<xs:element name="SecurityAudit" type="SecurityAuditType"/>
<xs:element name="SwitchAudit" type="ExtensionOnlyType"/>
<xs:element name="DeactivationCallCenter" type="ExtensionOnlyType"/>

<xs:element name="GoverningAgreements" type="GoverningAgreementsType">
  <xs:annotation>
    <xs:documentation>
      Provides a mechanism for linking to external (likely
      human readable) documents in which additional business agreements,
      (e.g. liability constraints, obligations, etc.) can be placed.
```

```
      </xs:documentation>
    </xs:annotation>
  </xs:element>

  <xs:element name="GoverningAgreementRef" type="GoverningAgreementRefType"/>

  <xs:simpleType name="nymType">
    <xs:restriction base="xs:NMTOKEN">
      <xs:enumeration value="anonymity"/>
      <xs:enumeration value="verinymity"/>
      <xs:enumeration value="pseudonymity"/>
    </xs:restriction>
  </xs:simpleType>

  <xs:complexType name="AuthnContextDeclarationBaseType">
    <xs:sequence>
      <xs:element ref="Identification" minOccurs="0"/>
      <xs:element ref="TechnicalProtection" minOccurs="0"/>
      <xs:element ref="OperationalProtection" minOccurs="0"/>
      <xs:element ref="AuthnMethod" minOccurs="0"/>
      <xs:element ref="GoverningAgreements" minOccurs="0"/>
      <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:attribute name="ID" type="xs:ID" use="optional"/>
  </xs:complexType>

  <xs:complexType name="IdentificationType">
    <xs:sequence>
      <xs:element ref="PhysicalVerification" minOccurs="0"/>
      <xs:element ref="WrittenConsent" minOccurs="0"/>
      <xs:element ref="GoverningAgreements" minOccurs="0"/>
      <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:attribute name="nym" type="nymType">
      <xs:annotation>
        <xs:documentation>
          This attribute indicates whether or not the
          Identification mechanisms allow the actions of the Principal to be
          linked to an actual end user.
        </xs:documentation>
      </xs:annotation>
    </xs:attribute>
  </xs:complexType>

  <xs:complexType name="TechnicalProtectionBaseType">
    <xs:sequence>
      <xs:choice minOccurs="0">
        <xs:element ref="PrivateKeyProtection"/>
        <xs:element ref="SecretKeyProtection"/>
      </xs:choice>
      <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>

  <xs:complexType name="OperationalProtectionType">
    <xs:sequence>
      <xs:element ref="SecurityAudit" minOccurs="0"/>
      <xs:element ref="DeactivationCallCenter" minOccurs="0"/>
      <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>

  <xs:complexType name="AuthnMethodBaseType">
    <xs:sequence>
      <xs:element ref="PrincipalAuthenticationMechanism" minOccurs="0"/>
      <xs:element ref="Authenticator" minOccurs="0"/>
      <xs:element ref="AuthenticatorTransportProtocol" minOccurs="0"/>
      <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
```

```
<xs:complexType name="GoverningAgreementsType">
  <xs:sequence>
    <xs:element ref="GoverningAgreementRef" maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>

<xs:complexType name="GoverningAgreementRefType">
  <xs:attribute name="governingAgreementRef" type="xs:anyURI" use="required"/>
</xs:complexType>

<xs:complexType name="PrincipalAuthenticationMechanismType">
  <xs:sequence>
    <xs:element ref="Password" minOccurs="0"/>
    <xs:element ref="RestrictedPassword" minOccurs="0"/>
    <xs:element ref="Token" minOccurs="0"/>
    <xs:element ref="Smartcard" minOccurs="0"/>
    <xs:element ref="ActivationPin" minOccurs="0"/>
    <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
  </xs:sequence>
  <xs:attribute name="preauth" type="xs:integer" use="optional"/>
</xs:complexType>

<xs:group name="AuthenticatorChoiceGroup">
  <xs:choice>
    <xs:element ref="PreviousSession"/>
    <xs:element ref="ResumeSession"/>
    <xs:element ref="DigSig"/>
    <xs:element ref="Password"/>
    <xs:element ref="RestrictedPassword"/>
    <xs:element ref="ZeroKnowledge"/>
    <xs:element ref="SharedSecretChallengeResponse"/>
    <xs:element ref="SharedSecretDynamicPlaintext"/>
    <xs:element ref="IPAddress"/>
    <xs:element ref="AsymmetricDecryption"/>
    <xs:element ref="AsymmetricKeyAgreement"/>
    <xs:element ref="SubscriberLineNumber"/>
    <xs:element ref="UserSuffix"/>
    <xs:element ref="ComplexAuthenticator"/>
  </xs:choice>
</xs:group>

<xs:group name="AuthenticatorSequenceGroup">
  <xs:sequence>
    <xs:element ref="PreviousSession" minOccurs="0"/>
    <xs:element ref="ResumeSession" minOccurs="0"/>
    <xs:element ref="DigSig" minOccurs="0"/>
    <xs:element ref="Password" minOccurs="0"/>
    <xs:element ref="RestrictedPassword" minOccurs="0"/>
    <xs:element ref="ZeroKnowledge" minOccurs="0"/>
    <xs:element ref="SharedSecretChallengeResponse" minOccurs="0"/>
    <xs:element ref="SharedSecretDynamicPlaintext" minOccurs="0"/>
    <xs:element ref="IPAddress" minOccurs="0"/>
    <xs:element ref="AsymmetricDecryption" minOccurs="0"/>
    <xs:element ref="AsymmetricKeyAgreement" minOccurs="0"/>
    <xs:element ref="SubscriberLineNumber" minOccurs="0"/>
    <xs:element ref="UserSuffix" minOccurs="0"/>
    <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
  </xs:sequence>
</xs:group>

<xs:complexType name="AuthenticatorBaseType">
  <xs:sequence>
    <xs:group ref="AuthenticatorChoiceGroup"/>
    <xs:group ref="AuthenticatorSequenceGroup"/>
  </xs:sequence>
</xs:complexType>

<xs:complexType name="ComplexAuthenticatorType">
  <xs:sequence>
    <xs:group ref="AuthenticatorChoiceGroup"/>
```

```
      <xs:group ref="AuthenticatorSequenceGroup"/>
    </xs:sequence>
  </xs:complexType>

  <xs:complexType name="AuthenticatorTransportProtocolType">
    <xs:sequence>
      <xs:choice minOccurs="0">
        <xs:element ref="HTTP"/>
        <xs:element ref="SSL"/>
        <xs:element ref="MobileNetworkNoEncryption"/>
        <xs:element ref="MobileNetworkRadioEncryption"/>
        <xs:element ref="MobileNetworkEndToEndEncryption"/>
        <xs:element ref="WTLS"/>
        <xs:element ref="IPSec"/>
        <xs:element ref="PSTN"/>
        <xs:element ref="ISDN"/>
        <xs:element ref="ADSL"/>
      </xs:choice>
      <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>

  <xs:complexType name="KeyActivationType">
    <xs:sequence>
      <xs:element ref="ActivationPin" minOccurs="0"/>
      <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>

  <xs:complexType name="KeySharingType">
    <xs:attribute name="sharing" type="xs:boolean" use="required"/>
  </xs:complexType>

  <xs:complexType name="PrivateKeyProtectionType">
    <xs:sequence>
      <xs:element ref="KeyActivation" minOccurs="0"/>
      <xs:element ref="KeyStorage" minOccurs="0"/>
      <xs:element ref="KeySharing" minOccurs="0"/>
      <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>

  <xs:complexType name="PasswordType">
    <xs:sequence>
      <xs:element ref="Length" minOccurs="0"/>
      <xs:element ref="Alphabet" minOccurs="0"/>
      <xs:element ref="Generation" minOccurs="0"/>
      <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:attribute name="ExternalVerification" type="xs:anyURI" use="optional"/>
  </xs:complexType>

  <xs:element name="RestrictedPassword" type="RestrictedPasswordType"/>

  <xs:complexType name="RestrictedPasswordType">
    <xs:complexContent>
      <xs:restriction base="PasswordType">
        <xs:sequence>
          <xs:element name="Length" type="RestrictedLengthType" minOccurs="1"/>
          <xs:element ref="Generation" minOccurs="0"/>
          <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
        </xs:sequence>
        <xs:attribute name="ExternalVerification" type="xs:anyURI"
use="optional"/>
      </xs:restriction>
    </xs:complexContent>
  </xs:complexType>

  <xs:complexType name="RestrictedLengthType">
    <xs:complexContent>
      <xs:restriction base="LengthType">
```

```
      <xs:attribute name="min" use="required">
        <xs:simpleType>
          <xs:restriction base="xs:integer">
            <xs:minInclusive value="3"/>
          </xs:restriction>
        </xs:simpleType>
      </xs:attribute>
      <xs:attribute name="max" type="xs:integer" use="optional"/>
    </xs:restriction>
  </xs:complexContent>
</xs:complexType>

<xs:complexType name="ActivationPinType">
  <xs:sequence>
    <xs:element ref="Length" minOccurs="0"/>
    <xs:element ref="Alphabet" minOccurs="0"/>
    <xs:element ref="Generation" minOccurs="0"/>
    <xs:element ref="ActivationLimit" minOccurs="0"/>
    <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>

<xs:element name="Alphabet" type="AlphabetType"/>
<xs:complexType name="AlphabetType">
  <xs:attribute name="requiredChars" type="xs:string" use="required"/>
  <xs:attribute name="excludedChars" type="xs:string" use="optional"/>
  <xs:attribute name="case" type="xs:string" use="optional"/>
</xs:complexType>

<xs:complexType name="TokenType">
  <xs:sequence>
    <xs:element ref="TimeSyncToken"/>
    <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>

<xs:simpleType name="DeviceTypeType">
  <xs:restriction base="xs:NMTOKEN">
    <xs:enumeration value="hardware"/>
    <xs:enumeration value="software"/>
  </xs:restriction>
</xs:simpleType>

<xs:simpleType name="booleanType">
  <xs:restriction base="xs:NMTOKEN">
    <xs:enumeration value="true"/>
    <xs:enumeration value="false"/>
  </xs:restriction>
</xs:simpleType>

<xs:complexType name="TimeSyncTokenType">
  <xs:attribute name="DeviceType" type="DeviceTypeType" use="required"/>
  <xs:attribute name="SeedLength" type="xs:integer" use="required"/>
  <xs:attribute name="DeviceInHand" type="booleanType" use="required"/>
</xs:complexType>

<xs:complexType name="ActivationLimitType">
  <xs:choice>
    <xs:element ref="ActivationLimitDuration"/>
    <xs:element ref="ActivationLimitUsages"/>
    <xs:element ref="ActivationLimitSession"/>
  </xs:choice>
</xs:complexType>

<xs:element name="ActivationLimitDuration" type="ActivationLimitDurationType">
  <xs:annotation>
    <xs:documentation>
      This element indicates u116 that the Key Activation Limit is
      defined as a specific duration of time.
    </xs:documentation>
  </xs:annotation>
```

```
  </xs:element>
```

```
  <xs:element name="ActivationLimitUsages" type="ActivationLimitUsagesType">
    <xs:annotation>
      <xs:documentation>
        This element indicates u116 that the Key Activation Limit is
        defined as a number of u117 usages.
      </xs:documentation>
    </xs:annotation>
  </xs:element>

  <xs:element name="ActivationLimitSession" type="ActivationLimitSessionType">
    <xs:annotation>
      <xs:documentation>
        This element indicates u116 that the Key Activation Limit is
        the session.
      </xs:documentation>
    </xs:annotation>
  </xs:element>

  <xs:complexType name="ActivationLimitDurationType">
    <xs:attribute name="duration" type="xs:duration" use="required"/>
  </xs:complexType>

  <xs:complexType name="ActivationLimitUsagesType">
    <xs:attribute name="number" type="xs:integer" use="required"/>
  </xs:complexType>

  <xs:complexType name="ActivationLimitSessionType"/>

  <xs:complexType name="LengthType">
    <xs:attribute name="min" type="xs:integer" use="required"/>
    <xs:attribute name="max" type="xs:integer" use="optional"/>
  </xs:complexType>

  <xs:simpleType name="mediumType">
    <xs:restriction base="xs:NMTOKEN">
      <xs:enumeration value="memory"/>
      <xs:enumeration value="smartcard"/>
      <xs:enumeration value="token"/>
      <xs:enumeration value="MobileDevice"/>
      <xs:enumeration value="MobileAuthCard"/>
    </xs:restriction>
  </xs:simpleType>

  <xs:complexType name="KeyStorageType">
    <xs:attribute name="medium" type="mediumType" use="required"/>
  </xs:complexType>

  <xs:complexType name="SecretKeyProtectionType">
    <xs:sequence>
      <xs:element ref="KeyActivation" minOccurs="0"/>
      <xs:element ref="KeyStorage" minOccurs="0"/>
      <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>

  <xs:complexType name="SecurityAuditType">
    <xs:sequence>
      <xs:element ref="SwitchAudit" minOccurs="0"/>
      <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>

  <xs:complexType name="ExtensionOnlyType">
    <xs:sequence>
      <xs:element ref="Extension" minOccurs="0"  maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>

  <xs:element name="Extension" type="ExtensionType"/>
```

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

</xs:schema>
```

## A.25 Схема SAML – Контекст аутентификации для X.509

Это Схема SAML Правил аутентификации для X.509.

```
<?xml version="1.0" encoding="UTF-8"?>

<xs:schema targetNamespace="urn:oasis:names:tc:SAML:2.0:ac:classes:X509"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns="urn:oasis:names:tc:SAML:2.0:ac:classes:X509"
  finalDefault="extension"
  blockDefault="substitution"
  version="2.0">

  <xs:redefine schemaLocation="saml-schema-authn-context-types-2.0.xsd">

    <xs:annotation>
      <xs:documentation>
        Class identifier: urn:oasis:names:tc:SAML:2.0:ac:classes:X509
        Document identifier: saml-schema-authn-context-x509-2.0
        Location: http://docs.oasis-open.org/security/saml/v2.0/
        Revision history:
          V2.0 (March, 2005):
            New authentication_u99 context class schema for SAML V2.0 .
      </xs:documentation>
    </xs:annotation>

    <xs:complexType name="AuthnContextDeclarationBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthnContextDeclarationBaseType">
          <xs:sequence>
            <xs:element ref="Identification" minOccurs="0"/>
            <xs:element ref="TechnicalProtection" minOccurs="0"/>
            <xs:element ref="OperationalProtection" minOccurs="0"/>
            <xs:element ref="AuthnMethod"/>
            <xs:element ref="GoverningAgreements" minOccurs="0"/>
            <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
          </xs:sequence>
          <xs:attribute name="ID" type="xs:ID" use="optional"/>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="AuthnMethodBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthnMethodBaseType">
          <xs:sequence>
            <xs:element ref="PrincipalAuthenticationMechanism"/>
            <xs:element ref="Authenticator"/>
            <xs:element ref="AuthenticatorTransportProtocol" minOccurs="0"/>
            <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="PrincipalAuthenticationMechanismType">
      <xs:complexContent>
        <xs:restriction base="PrincipalAuthenticationMechanismType">
          <xs:sequence>
            <xs:element ref="RestrictedPassword"/>
          </xs:sequence>
```

```
          <xs:attribute name="preauth" type="xs:integer" use="optional"/>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="AuthenticatorBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthenticatorBaseType">
          <xs:sequence>
            <xs:element ref="DigSig"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="PublicKeyType">
      <xs:complexContent>
        <xs:restriction base="PublicKeyType">
          <xs:attribute name="keyValidation" type="xs:anyURI"
fixed="urn:oasis:names:tc:SAML:2.0:ac:classes:X509"/>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

  </xs:redefine>

</xs:schema>
```

## A.26 Схема SAML – Контекст аутентификации для XMLDSig

Это Схема SAML Правил аутентификации для цифровой подписи XML.

```
<?xml version="1.0" encoding="UTF-8"?>

<xs:schema targetNamespace="urn:oasis:names:tc:SAML:2.0:ac:classes:XMLDSig"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns="urn:oasis:names:tc:SAML:2.0:ac:classes:XMLDSig"
  finalDefault="extension"
  blockDefault="substitution"
  version="2.0">

  <xs:redefine schemaLocation="saml-schema-authn-context-types-2.0.xsd">

    <xs:annotation>
      <xs:documentation>
        Class identifier: urn:oasis:names:tc:SAML:2.0:ac:classes:XMLDSig
        Document identifier: saml-schema-authn-context-xmldsig-2.0
        Location: http://docs.oasis-open.org/security/saml/v2.0/
        Revision history:
          V2.0 (March, 2005):
            New authentication_u99 context class schema for SAML V2.0 .
      </xs:documentation>
    </xs:annotation>

    <xs:complexType name="AuthnContextDeclarationBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthnContextDeclarationBaseType">
          <xs:sequence>
            <xs:element ref="Identification" minOccurs="0"/>
            <xs:element ref="TechnicalProtection" minOccurs="0"/>
            <xs:element ref="OperationalProtection" minOccurs="0"/>
            <xs:element ref="AuthnMethod"/>
            <xs:element ref="GoverningAgreements" minOccurs="0"/>
            <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
          </xs:sequence>
          <xs:attribute name="ID" type="xs:ID" use="optional"/>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>
```

```
    <xs:complexType name="AuthnMethodBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthnMethodBaseType">
          <xs:sequence>
            <xs:element ref="PrincipalAuthenticationMechanism"/>
            <xs:element ref="Authenticator"/>
            <xs:element ref="AuthenticatorTransportProtocol" minOccurs="0"/>
            <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="PrincipalAuthenticationMechanismType">
      <xs:complexContent>
        <xs:restriction base="PrincipalAuthenticationMechanismType">
          <xs:sequence>
            <xs:element ref="RestrictedPassword"/>
          </xs:sequence>
          <xs:attribute name="preauth" type="xs:integer" use="optional"/>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="AuthenticatorBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthenticatorBaseType">
          <xs:sequence>
            <xs:element ref="DigSig"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="PublicKeyType">
      <xs:complexContent>
        <xs:restriction base="PublicKeyType">
          <xs:attribute name="keyValidation" type="xs:anyURI"
fixed="urn:ietf:rfc:3075"/>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

  </xs:redefine>

</xs:schema>
```

## A.27 Схема SAML ECP

Это листинг Схемы SAML профиля с расширенным клиентом или прокси-сервером (ECP).

```
<?xml version="1.0" encoding="UTF-8"?>
<schema
    targetNamespace="urn:oasis:names:tc:SAML:2.0:profiles:SSO:ecp"
    xmlns="http://www.w3.org/2001/XMLSchema"
    xmlns:ecp="urn:oasis:names:tc:SAML:2.0:profiles:SSO:ecp"
    xmlns:samlp="urn:oasis:names:tc:SAML:2.0:protocol"
    xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion"
    xmlns:S="http://schemas.xmlsoap.org/soap/envelope/"
    elementFormDefault="unqualified"
    attributeFormDefault="unqualified"
    blockDefault="substitution"
    version="2.0">
    <import namespace="urn:oasis:names:tc:SAML:2.0:protocol"
        schemaLocation="saml-schema-protocol-2.0.xsd"/>
    <import namespace="urn:oasis:names:tc:SAML:2.0:assertion"
        schemaLocation="saml-schema-assertion-2.0.xsd"/>
    <import namespace="http://schemas.xmlsoap.org/soap/envelope/"
        schemaLocation="http://schemas.xmlsoap.org/soap/envelope/"/>
```

```
    <annotation>
        <documentation>
            Document identifier: saml-schema-ecp-2.0
            Location: http://docs.oasis-open.org/security/saml/v2.0/
            Revision history:
              V2.0 (March, 2005):
                Custom schema for ECP profile, first published in SAML 2.0.
        </documentation>
    </annotation>

    <element name="Request" type="ecp:RequestType"/>
    <complexType name="RequestType">
        <sequence>
            <element ref="saml:Issuer"/>
            <element ref="samlp:IDPList" minOccurs="0"/>
        </sequence>
        <attribute ref="S:mustUnderstand" use="required"/>
        <attribute ref="S:actor" use="required"/>
        <attribute name="ProviderName" type="string" use="optional"/>
        <attribute name="IsPassive" type="boolean" use="optional"/>
    </complexType>

    <element name="Response" type="ecp:ResponseType"/>
    <complexType name="ResponseType">
        <attribute ref="S:mustUnderstand" use="required"/>
        <attribute ref="S:actor" use="required"/>
        <attribute name="AssertionConsumerServiceURL" type="anyURI"
use="required"/>
    </complexType>

    <element name="RelayState" type="ecp:RelayStateType"/>
    <complexType name="RelayStateType">
        <simpleContent>
            <extension base="string">
                <attribute ref="S:mustUnderstand" use="required"/>
                <attribute ref="S:actor" use="required"/>
            </extension>
        </simpleContent>
    </complexType>
</schema>
```

## A.28 Схема SAML для метаданных

Это листинг Схемы SAML для метаданных.

```
<?xml version="1.0" encoding="UTF-8"?>
<schema
    targetNamespace="urn:oasis:names:tc:SAML:2.0:metadata"
    xmlns:md="urn:oasis:names:tc:SAML:2.0:metadata"
    xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
    xmlns:xenc="http://www.w3.org/2001/04/xmlenc#"
    xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion"
    xmlns="http://www.w3.org/2001/XMLSchema"
    elementFormDefault="unqualified"
    attributeFormDefault="unqualified"
    blockDefault="substitution"
    version="2.0">
    <import namespace="http://www.w3.org/2000/09/xmldsig#"
        schemaLocation="http://www.w3.org/TR/2002/REC-xmldsig-core-
20020212/xmldsig-core-schema.xsd"/>
    <import namespace="http://www.w3.org/2001/04/xmlenc#"
        schemaLocation="http://www.w3.org/TR/2002/REC-xmlenc-core-20021210/xenc-
schema.xsd"/>
    <import namespace="urn:oasis:names:tc:SAML:2.0:assertion"
        schemaLocation="saml-schema-assertion-2.0.xsd"/>
    <import namespace="http://www.w3.org/XML/1998/namespace"
        schemaLocation="http://www.w3.org/2001/xml.xsd"/>
    <annotation>
        <documentation>
```

```
            Document identifier: saml-schema-metadata-2.0
            Location: http://docs.oasis-open.org/security/saml/v2.0/
            Revision history:
              V2.0 (March, 2005):
                Schema for Metadata SAML, first published in SAML 2.0.
        </documentation>
    </annotation>

    <simpleType name="entityIDType">
        <restriction base="anyURI">
            <maxLength value="1024"/>
        </restriction>
    </simpleType>
    <complexType name="localizedNameType">
        <simpleContent>
            <extension base="string">
                <attribute ref="xml:lang" use="required"/>
            </extension>
        </simpleContent>
    </complexType>
    <complexType name="localizedURIType">
        <simpleContent>
            <extension base="anyURI">
                <attribute ref="xml:lang" use="required"/>
            </extension>
        </simpleContent>
    </complexType>

    <element name="Extensions" type="md:ExtensionsType"/>
    <complexType final="#all" name="ExtensionsType">
        <sequence>
            <any namespace="##other" processContents="lax" maxOccurs="unbounded"/>
        </sequence>
    </complexType>

    <complexType name="EndpointType">
        <sequence>
            <any namespace="##other" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
        </sequence>
        <attribute name="Binding" type="anyURI" use="required"/>
        <attribute name="Location" type="anyURI" use="required"/>
        <attribute name="ResponseLocation" type="anyURI" use="optional"/>
        <anyAttribute namespace="##other" processContents="lax"/>
    </complexType>

    <complexType name="IndexedEndpointType">
        <complexContent>
            <extension base="md:EndpointType">
                <attribute name="index" type="unsignedShort" use="required"/>
                <attribute name="isDefault" type="boolean" use="optional"/>
            </extension>
        </complexContent>
    </complexType>

    <element name="EntitiesDescriptor" type="md:EntitiesDescriptorType"/>
    <complexType name="EntitiesDescriptorType">
        <sequence>
            <element ref="ds:Signature" minOccurs="0"/>
            <element ref="md:Extensions" minOccurs="0"/>
            <choice minOccurs="1" maxOccurs="unbounded">
                <element ref="md:EntityDescriptor"/>
                <element ref="md:EntitiesDescriptor"/>
            </choice>
        </sequence>
        <attribute name="validUntil" type="dateTime" use="optional"/>
        <attribute name="cacheDuration" type="duration" use="optional"/>
        <attribute name="ID" type="ID" use="optional"/>
        <attribute name="Name" type="string" use="optional"/>
    </complexType>
```

```
    <element name="EntityDescriptor" type="md:EntityDescriptorType"/>
    <complexType name="EntityDescriptorType">
        <sequence>
            <element ref="ds:Signature" minOccurs="0"/>
            <element ref="md:Extensions" minOccurs="0"/>
            <choice>
                <choice maxOccurs="unbounded">
                    <element ref="md:RoleDescriptor"/>
                    <element ref="md:IDPSSODescriptor"/>
                    <element ref="md:SPSSODescriptor"/>
                    <element ref="md:AuthnAuthorityDescriptor"/>
                    <element ref="md:AttributeAuthorityDescriptor"/>
                    <element ref="md:PDPDescriptor"/>
                </choice>
                <element ref="md:AffiliationDescriptor"/>
            </choice>
            <element ref="md:Organization" minOccurs="0"/>
            <element ref="md:ContactPerson" minOccurs="0" maxOccurs="unbounded"/>
            <element ref="md:AdditionalMetadataLocation" minOccurs="0"
maxOccurs="unbounded"/>
        </sequence>
        <attribute name="entityID" type="md:entityIDType" use="required"/>
        <attribute name="validUntil" type="dateTime" use="optional"/>
        <attribute name="cacheDuration" type="duration" use="optional"/>
        <attribute name="ID" type="ID" use="optional"/>
        <anyAttribute namespace="##other" processContents="lax"/>
    </complexType>

    <element name="Organization" type="md:OrganizationType"/>
    <complexType name="OrganizationType">
        <sequence>
            <element ref="md:Extensions" minOccurs="0"/>
            <element ref="md:OrganizationName" maxOccurs="unbounded"/>
            <element ref="md:OrganizationDisplayName" maxOccurs="unbounded"/>
            <element ref="md:OrganizationURL" maxOccurs="unbounded"/>
        </sequence>
        <anyAttribute namespace="##other" processContents="lax"/>
    </complexType>
    <element name="OrganizationName" type="md:localizedNameType"/>
    <element name="OrganizationDisplayName" type="md:localizedNameType"/>
    <element name="OrganizationURL" type="md:localizedURIType"/>
    <element name="ContactPerson" type="md:ContactType"/>
    <complexType name="ContactType">
        <sequence>
            <element ref="md:Extensions" minOccurs="0"/>
            <element ref="md:Company" minOccurs="0"/>
            <element ref="md:GivenName" minOccurs="0"/>
            <element ref="md:SurName" minOccurs="0"/>
            <element ref="md:EmailAddress" minOccurs="0" maxOccurs="unbounded"/>
            <element ref="md:TelephoneNumber" minOccurs="0"
maxOccurs="unbounded"/>
        </sequence>
        <attribute name="contactType" type="md:ContactTypeType" use="required"/>
        <anyAttribute namespace="##other" processContents="lax"/>
    </complexType>
    <element name="Company" type="string"/>
    <element name="GivenName" type="string"/>
    <element name="SurName" type="string"/>
    <element name="EmailAddress" type="anyURI"/>
    <element name="TelephoneNumber" type="string"/>
    <simpleType name="ContactTypeType">
        <restriction base="string">
            <enumeration value="technical"/>
            <enumeration value="support"/>
            <enumeration value="administrative"/>
            <enumeration value="billing"/>
            <enumeration value="other"/>
        </restriction>
    </simpleType>
```

```
    <element name="AdditionalMetadataLocation"
type="md:AdditionalMetadataLocationType"/>
    <complexType name="AdditionalMetadataLocationType">
        <simpleContent>
            <extension base="anyURI">
                <attribute name="namespace" type="anyURI" use="required"/>
            </extension>
        </simpleContent>
    </complexType>

    <element name="RoleDescriptor" type="md:RoleDescriptorType"/>
    <complexType name="RoleDescriptorType" abstract="true">
        <sequence>
            <element ref="ds:Signature" minOccurs="0"/>
            <element ref="md:Extensions" minOccurs="0"/>
            <element ref="md:KeyDescriptor" minOccurs="0" maxOccurs="unbounded"/>
            <element ref="md:Organization" minOccurs="0"/>
            <element ref="md:ContactPerson" minOccurs="0" maxOccurs="unbounded"/>
        </sequence>
        <attribute name="ID" type="ID" use="optional"/>
        <attribute name="validUntil" type="dateTime" use="optional"/>
        <attribute name="cacheDuration" type="duration" use="optional"/>
        <attribute name="protocolSupportEnumeration" type="md:anyURIListType"
use="required"/>
        <attribute name="errorURL" type="anyURI" use="optional"/>
        <anyAttribute namespace="##other" processContents="lax"/>
    </complexType>
    <simpleType name="anyURIListType">
        <list itemType="anyURI"/>
    </simpleType>

    <element name="KeyDescriptor" type="md:KeyDescriptorType"/>
    <complexType name="KeyDescriptorType">
        <sequence>
            <element ref="ds:KeyInfo"/>
            <element ref="md:EncryptionMethod" minOccurs="0"
maxOccurs="unbounded"/>
        </sequence>
        <attribute name="use" type="md:KeyTypes" use="optional"/>
    </complexType>
    <simpleType name="KeyTypes">
        <restriction base="string">
            <enumeration value="encryption"/>
            <enumeration value="signing"/>
        </restriction>
    </simpleType>
    <element name="EncryptionMethod" type="xenc:EncryptionMethodType"/>

    <complexType name="SSODescriptorType" abstract="true">
        <complexContent>
            <extension base="md:RoleDescriptorType">
                <sequence>
                    <element ref="md:ArtifactResolutionService" minOccurs="0"
maxOccurs="unbounded"/>
                    <element ref="md:SingleLogoutService" minOccurs="0"
maxOccurs="unbounded"/>
                    <element ref="md:ManageNameIDService" minOccurs="0"
maxOccurs="unbounded"/>
                    <element ref="md:NameIDFormat" minOccurs="0"
maxOccurs="unbounded"/>
                </sequence>
            </extension>
        </complexContent>
    </complexType>
    <element name="ArtifactResolutionService" type="md:IndexedEndpointType"/>
    <element name="SingleLogoutService" type="md:EndpointType"/>
    <element name="ManageNameIDService" type="md:EndpointType"/>
    <element name="NameIDFormat" type="anyURI"/>

    <element name="IDPSSODescriptor" type="md:IDPSSODescriptorType"/>
```

```
    <complexType name="IDPSSODescriptorType">
        <complexContent>
            <extension base="md:SSODescriptorType">
                <sequence>
                    <element ref="md:SingleSignOnService" maxOccurs="unbounded"/>
                    <element ref="md:NameIDMappingService" minOccurs="0"
maxOccurs="unbounded"/>
                    <element ref="md:AssertionIDRequestService" minOccurs="0"
maxOccurs="unbounded"/>
                    <element ref="md:AttributeProfile" minOccurs="0"
maxOccurs="unbounded"/>
                    <element ref="saml:Attribute" minOccurs="0"
maxOccurs="unbounded"/>
                </sequence>
                <attribute name="WantAuthnRequestsSigned" type="boolean"
use="optional"/>
            </extension>
        </complexContent>
    </complexType>
    <element name="SingleSignOnService" type="md:EndpointType"/>
    <element name="NameIDMappingService" type="md:EndpointType"/>
    <element name="AssertionIDRequestService" type="md:EndpointType"/>
    <element name="AttributeProfile" type="anyURI"/>

    <element name="SPSSODescriptor" type="md:SPSSODescriptorType"/>
    <complexType name="SPSSODescriptorType">
        <complexContent>
            <extension base="md:SSODescriptorType">
                <sequence>
                    <element ref="md:AssertionConsumerService"
maxOccurs="unbounded"/>
                    <element ref="md:AttributeConsumingService" minOccurs="0"
maxOccurs="unbounded"/>
                </sequence>
                <attribute name="AuthnRequestsSigned" type="boolean"
use="optional"/>
                <attribute name="WantAssertionsSigned" type="boolean"
use="optional"/>
            </extension>
        </complexContent>
    </complexType>
    <element name="AssertionConsumerService" type="md:IndexedEndpointType"/>
    <element name="AttributeConsumingService"
type="md:AttributeConsumingServiceType"/>
    <complexType name="AttributeConsumingServiceType">
        <sequence>
            <element ref="md:ServiceName" maxOccurs="unbounded"/>
            <element ref="md:ServiceDescription" minOccurs="0"
maxOccurs="unbounded"/>
            <element ref="md:RequestedAttribute" maxOccurs="unbounded"/>
        </sequence>
        <attribute name="index" type="unsignedShort" use="required"/>
        <attribute name="isDefault" type="boolean" use="optional"/>
    </complexType>
    <element name="ServiceName" type="md:localizedNameType"/>
    <element name="ServiceDescription" type="md:localizedNameType"/>
    <element name="RequestedAttribute" type="md:RequestedAttributeType"/>
    <complexType name="RequestedAttributeType">
        <complexContent>
            <extension base="saml:AttributeType">
                <attribute name="isRequired" type="boolean" use="optional"/>
            </extension>
        </complexContent>
    </complexType>

    <element name="AuthnAuthorityDescriptor"
type="md:AuthnAuthorityDescriptorType"/>
    <complexType name="AuthnAuthorityDescriptorType">
        <complexContent>
            <extension base="md:RoleDescriptorType">
                <sequence>
```

```
                    <element ref="md:AuthnQueryService" maxOccurs="unbounded"/>
                    <element ref="md:AssertionIDRequestService" minOccurs="0"
maxOccurs="unbounded"/>
                    <element ref="md:NameIDFormat" minOccurs="0"
maxOccurs="unbounded"/>
                </sequence>
            </extension>
        </complexContent>
    </complexType>
    <element name="AuthnQueryService" type="md:EndpointType"/>

    <element name="PDPDescriptor" type="md:PDPDescriptorType"/>
    <complexType name="PDPDescriptorType">
        <complexContent>
            <extension base="md:RoleDescriptorType">
                <sequence>
                    <element ref="md:AuthzService" maxOccurs="unbounded"/>
                    <element ref="md:AssertionIDRequestService" minOccurs="0"
maxOccurs="unbounded"/>
                    <element ref="md:NameIDFormat" minOccurs="0"
maxOccurs="unbounded"/>
                </sequence>
            </extension>
        </complexContent>
    </complexType>
    <element name="AuthzService" type="md:EndpointType"/>

    <element name="AttributeAuthorityDescriptor"
type="md:AttributeAuthorityDescriptorType"/>
    <complexType name="AttributeAuthorityDescriptorType">
        <complexContent>
            <extension base="md:RoleDescriptorType">
                <sequence>
                    <element ref="md:AttributeService" maxOccurs="unbounded"/>
                    <element ref="md:AssertionIDRequestService" minOccurs="0"
maxOccurs="unbounded"/>
                    <element ref="md:NameIDFormat" minOccurs="0"
maxOccurs="unbounded"/>
                    <element ref="md:AttributeProfile" minOccurs="0"
maxOccurs="unbounded"/>
                    <element ref="saml:Attribute" minOccurs="0"
maxOccurs="unbounded"/>
                </sequence>
            </extension>
        </complexContent>
    </complexType>
    <element name="AttributeService" type="md:EndpointType"/>

    <element name="AffiliationDescriptor" type="md:AffiliationDescriptorType"/>
    <complexType name="AffiliationDescriptorType">
        <sequence>
            <element ref="ds:Signature" minOccurs="0"/>
            <element ref="md:Extensions" minOccurs="0"/>
            <element ref="md:AffiliateMember" maxOccurs="unbounded"/>
            <element ref="md:KeyDescriptor" minOccurs="0" maxOccurs="unbounded"/>
        </sequence>
        <attribute name="affiliationOwnerID" type="md:entityIDType"
use="required"/>
        <attribute name="validUntil" type="dateTime" use="optional"/>
        <attribute name="cacheDuration" type="duration" use="optional"/>
        <attribute name="ID" type="ID" use="optional"/>
        <anyAttribute namespace="##other" processContents="lax"/>
    </complexType>
    <element name="AffiliateMember" type="md:entityIDType"/>
</schema>
```

### A.29 Схема SAML – Протокол

Это листинг Схемы SAML для протокола SAML.

```
<?xml version="1.0" encoding="UTF-8"?>
<schema
    targetNamespace="urn:oasis:names:tc:SAML:2.0:protocol"
    xmlns="http://www.w3.org/2001/XMLSchema"
    xmlns:samlp="urn:oasis:names:tc:SAML:2.0:protocol"
    xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion"
    xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
    elementFormDefault="unqualified"
    attributeFormDefault="unqualified"
    blockDefault="substitution"
    version="2.0">
    <import namespace="urn:oasis:names:tc:SAML:2.0:assertion"
        schemaLocation="saml-schema-assertion-2.0.xsd"/>
    <import namespace="http://www.w3.org/2000/09/xmldsig#"
        schemaLocation="http://www.w3.org/TR/2002/REC-xmldsig-core-
20020212/xmldsig-core-schema.xsd"/>
    <annotation>
        <documentation>
            Document identifier: saml-schema-protocol-2.0
            Location: http://docs.oasis-open.org/security/saml/v2.0/
            Revision history:
            V1.0 (November, 2002):
              Initial Standard u83 schema.
            V1.1 (September, 2003):
              Updates within the same V1.0 namespace.
            V2.0 (March, 2005):
              New protocol schema based in a SAML V2.0 namespace.
     </documentation>
    </annotation>
    <complexType name="RequestAbstractType" abstract="true">
        <sequence>
            <element ref="saml:Issuer" minOccurs="0"/>
            <element ref="ds:Signature" minOccurs="0"/>
            <element ref="samlp:Extensions" minOccurs="0"/>
        </sequence>
        <attribute name="ID" type="ID" use="required"/>
        <attribute name="Version" type="string" use="required"/>
        <attribute name="IssueInstant" type="dateTime" use="required"/>
        <attribute name="Destination" type="anyURI" use="optional"/>
      <attribute name="Consent" type="anyURI" use="optional"/>
    </complexType>
    <element name="Extensions" type="samlp:ExtensionsType"/>
    <complexType name="ExtensionsType">
        <sequence>
            <any namespace="##other" processContents="lax" maxOccurs="unbounded"/>
        </sequence>
    </complexType>
    <complexType name="StatusResponseType">
      <sequence>
            <element ref="saml:Issuer" minOccurs="0"/>
            <element ref="ds:Signature" minOccurs="0"/>
            <element ref="samlp:Extensions" minOccurs="0"/>
            <element ref="samlp:Status"/>
      </sequence>
      <attribute name="ID" type="ID" use="required"/>
      <attribute name="InResponseTo" type="NCName" use="optional"/>
      <attribute name="Version" type="string" use="required"/>
      <attribute name="IssueInstant" type="dateTime" use="required"/>
      <attribute name="Destination" type="anyURI" use="optional"/>
      <attribute name="Consent" type="anyURI" use="optional"/>
    </complexType>
    <element name="Status" type="samlp:StatusType"/>
    <complexType name="StatusType">
        <sequence>
            <element ref="samlp:StatusCode"/>
            <element ref="samlp:StatusMessage" minOccurs="0"/>
            <element ref="samlp:StatusDetail" minOccurs="0"/>
        </sequence>
```

```
    </complexType>
    <element name="StatusCode" type="samlp:StatusCodeType"/>
    <complexType name="StatusCodeType">
        <sequence>
            <element ref="samlp:StatusCode" minOccurs="0"/>
        </sequence>
        <attribute name="Value" type="anyURI" use="required"/>
    </complexType>
    <element name="StatusMessage" type="string"/>
    <element name="StatusDetail" type="samlp:StatusDetailType"/>
    <complexType name="StatusDetailType">
        <sequence>
            <any namespace="##any" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
        </sequence>
    </complexType>
    <element name="AssertionIDRequest" type="samlp:AssertionIDRequestType"/>
    <complexType name="AssertionIDRequestType">
      <complexContent>
            <extension base="samlp:RequestAbstractType">
                <sequence>
                    <element ref="saml:AssertionIDRef" maxOccurs="unbounded"/>
                </sequence>
            </extension>
      </complexContent>
    </complexType>
    <element name="SubjectQuery" type="samlp:SubjectQueryAbstractType"/>
    <complexType name="SubjectQueryAbstractType" abstract="true">
      <complexContent>
            <extension base="samlp:RequestAbstractType">
                <sequence>
                    <element ref="saml:Subject"/>
                </sequence>
            </extension>
      </complexContent>
    </complexType>
    <element name="AuthnQuery" type="samlp:AuthnQueryType"/>
    <complexType name="AuthnQueryType">
        <complexContent>
            <extension base="samlp:SubjectQueryAbstractType">
                <sequence>
                    <element ref="samlp:RequestedAuthnContext" minOccurs="0"/>
                </sequence>
                <attribute name="SessionIndex" type="string" use="optional"/>
            </extension>
        </complexContent>
    </complexType>
    <element name="RequestedAuthnContext" type="samlp:RequestedAuthnContextType"/>
    <complexType name="RequestedAuthnContextType">
        <choice>
            <element ref="saml:AuthnContextClassRef" maxOccurs="unbounded"/>
            <element ref="saml:AuthnContextDeclRef" maxOccurs="unbounded"/>
        </choice>
        <attribute name="Comparison" type="samlp:AuthnContextComparisonType"
use="optional"/>
    </complexType>
    <simpleType name="AuthnContextComparisonType">
        <restriction base="string">
            <enumeration value="exact"/>
            <enumeration value="minimum"/>
            <enumeration value="maximum"/>
            <enumeration value="better"/>
        </restriction>
    </simpleType>
    <element name="AttributeQuery" type="samlp:AttributeQueryType"/>
    <complexType name="AttributeQueryType">
        <complexContent>
            <extension base="samlp:SubjectQueryAbstractType">
                <sequence>
```

```
                    <element ref="saml:Attribute" minOccurs="0"
maxOccurs="unbounded"/>
                </sequence>
            </extension>
        </complexContent>
    </complexType>
    <element name="AuthzDecisionQuery" type="samlp:AuthzDecisionQueryType"/>
    <complexType name="AuthzDecisionQueryType">
        <complexContent>
            <extension base="samlp:SubjectQueryAbstractType">
                <sequence>
                    <element ref="saml:Action" maxOccurs="unbounded"/>
                    <element ref="saml:Evidence" minOccurs="0"/>
                </sequence>
                <attribute name="Resource" type="anyURI" use="required"/>
            </extension>
        </complexContent>
    </complexType>
    <element name="AuthnRequest" type="samlp:AuthnRequestType"/>
    <complexType name="AuthnRequestType">
        <complexContent>
            <extension base="samlp:RequestAbstractType">
                <sequence>
                    <element ref="saml:Subject" minOccurs="0"/>
                    <element ref="samlp:NameIDPolicy" minOccurs="0"/>
                    <element ref="saml:Conditions" minOccurs="0"/>
                    <element ref="samlp:RequestedAuthnContext" minOccurs="0"/>
                    <element ref="samlp:Scoping" minOccurs="0"/>
                </sequence>
                <attribute name="ForceAuthn" type="boolean" use="optional"/>
                <attribute name="IsPassive" type="boolean" use="optional"/>
                <attribute name="ProtocolBinding" type="anyURI" use="optional"/>
                <attribute name="AssertionConsumerServiceIndex"
type="unsignedShort" use="optional"/>
                <attribute name="AssertionConsumerServiceURL" type="anyURI"
use="optional"/>
                <attribute name="AttributeConsumingServiceIndex"
type="unsignedShort" use="optional"/>
                <attribute name="ProviderName" type="string" use="optional"/>
            </extension>
        </complexContent>
    </complexType>
    <element name="NameIDPolicy" type="samlp:NameIDPolicyType"/>
    <complexType name="NameIDPolicyType">
        <attribute name="Format" type="anyURI" use="optional"/>
        <attribute name="SPNameQualifier" type="string" use="optional"/>
        <attribute name="AllowCreate" type="boolean" use="optional"/>
    </complexType>
    <element name="Scoping" type="samlp:ScopingType"/>
    <complexType name="ScopingType">
        <sequence>
            <element ref="samlp:IDPList" minOccurs="0"/>
            <element ref="samlp:RequesterID" minOccurs="0" maxOccurs="unbounded"/>
        </sequence>
        <attribute name="ProxyCount" type="nonNegativeInteger" use="optional"/>
    </complexType>
    <element name="RequesterID" type="anyURI"/>
    <element name="IDPList" type="samlp:IDPListType"/>
    <complexType name="IDPListType">
        <sequence>
            <element ref="samlp:IDPEntry" maxOccurs="unbounded"/>
            <element ref="samlp:GetComplete" minOccurs="0"/>
        </sequence>
    </complexType>
    <element name="IDPEntry" type="samlp:IDPEntryType"/>
    <complexType name="IDPEntryType">
        <attribute name="ProviderID" type="anyURI" use="required"/>
        <attribute name="Name" type="string" use="optional"/>
        <attribute name="Loc" type="anyURI" use="optional"/>
    </complexType>
```

```
    <element name="GetComplete" type="anyURI"/>
    <element name="Response" type="samlp:ResponseType"/>
    <complexType name="ResponseType">
      <complexContent>
            <extension base="samlp:StatusResponseType">
                <choice minOccurs="0" maxOccurs="unbounded">
                    <element ref="saml:Assertion"/>
                    <element ref="saml:EncryptedAssertion"/>
                </choice>
            </extension>
      </complexContent>
    </complexType>
    <element name="ArtifactResolve" type="samlp:ArtifactResolveType"/>
    <complexType name="ArtifactResolveType">
      <complexContent>
            <extension base="samlp:RequestAbstractType">
                <sequence>
                    <element ref="samlp:Artifact"/>
                </sequence>
            </extension>
      </complexContent>
    </complexType>
    <element name="Artifact" type="string"/>
    <element name="ArtifactResponse" type="samlp:ArtifactResponseType"/>
    <complexType name="ArtifactResponseType">
      <complexContent>
            <extension base="samlp:StatusResponseType">
                <sequence>
                    <any namespace="##any" processContents="lax" minOccurs="0"/>
                </sequence>
            </extension>
      </complexContent>
    </complexType>
    <element name="ManageNameIDRequest" type="samlp:ManageNameIDRequestType"/>
    <complexType name="ManageNameIDRequestType">
      <complexContent>
            <extension base="samlp:RequestAbstractType">
                <sequence>
                    <choice>
                        <element ref="saml:NameID"/>
                        <element ref="saml:EncryptedID"/>
                    </choice>
                    <choice>
                        <element ref="samlp:NewID"/>
                        <element ref="samlp:NewEncryptedID"/>
                        <element ref="samlp:Terminate"/>
                    </choice>
                </sequence>
            </extension>
      </complexContent>
    </complexType>
    <element name="NewID" type="string"/>
    <element name="NewEncryptedID" type="saml:EncryptedElementType"/>
    <element name="Terminate" type="samlp:TerminateType"/>
    <complexType name="TerminateType"/>
    <element name="ManageNameIDResponse" type="samlp:StatusResponseType"/>
    <element name="LogoutRequest" type="samlp:LogoutRequestType"/>
    <complexType name="LogoutRequestType">
        <complexContent>
            <extension base="samlp:RequestAbstractType">
                <sequence>
                    <choice>
                        <element ref="saml:BaseID"/>
                        <element ref="saml:NameID"/>
                        <element ref="saml:EncryptedID"/>
                    </choice>
                    <element ref="samlp:SessionIndex" minOccurs="0"
maxOccurs="unbounded"/>
                </sequence>
                <attribute name="Reason" type="string" use="optional"/>
                <attribute name="NotOnOrAfter" type="dateTime" use="optional"/>
```

```
            </extension>
        </complexContent>
    </complexType>
    <element name="SessionIndex" type="string"/>
    <element name="LogoutResponse" type="samlp:StatusResponseType"/>
    <element name="NameIDMappingRequest" type="samlp:NameIDMappingRequestType"/>
    <complexType name="NameIDMappingRequestType">
        <complexContent>
            <extension base="samlp:RequestAbstractType">
                <sequence>
                    <choice>
                        <element ref="saml:BaseID"/>
                        <element ref="saml:NameID"/>
                        <element ref="saml:EncryptedID"/>
                    </choice>
                    <element ref="samlp:NameIDPolicy"/>
                </sequence>
            </extension>
        </complexContent>
    </complexType>
    <element name="NameIDMappingResponse" type="samlp:NameIDMappingResponseType"/>
    <complexType name="NameIDMappingResponseType">
        <complexContent>
            <extension base="samlp:StatusResponseType">
                <choice>
                    <element ref="saml:NameID"/>
                    <element ref="saml:EncryptedID"/>
                </choice>
            </extension>
        </complexContent>
    </complexType>
</schema>
```

## A.30 Схема SAML X.500

Это листинг Схемы SAML для X.500.

```
<?xml version="1.0" encoding="UTF-8"?>
<schema
    targetNamespace="urn:oasis:names:tc:SAML:2.0:profiles:attribute:X500"
    xmlns="http://www.w3.org/2001/XMLSchema"
    elementFormDefault="unqualified"
    attributeFormDefault="unqualified"
    blockDefault="substitution"
    version="2.0">
    <annotation>
        <documentation>
            Document identifier: saml-schema-x500-2.0
            Location: http://docs.oasis-open.org/security/saml/v2.0/
            Revision history:
              V2.0 (March, 2005):
                Custom schema for X.500 attribute profile, first published in SAML 2.0.
        </documentation>
    </annotation>
    <attribute name="Encoding" type="string"/>
</schema>
```

## A.31 Схема SAML XACML

Это листинг Схемы SAML для XACML.

```
<?xml version="1.0" encoding="UTF-8"?>
<schema
    targetNamespace="urn:oasis:names:tc:SAML:2.0:profiles:attribute:XACML"
    xmlns="http://www.w3.org/2001/XMLSchema"
    elementFormDefault="unqualified"
    attributeFormDefault="unqualified"
    blockDefault="substitution"
```

```
    version="2.0">
    <annotation>
        <documentation>
            Document identifier: saml-schema-xacml-2.0
            Location: http://docs.oasis-open.org/security/saml/v2.0/
            Revision history:
            V2.0 (March, 2005):
              Custom schema for XACML attribute profile, first published in SAML 2.0.
        </documentation>
    </annotation>
    <attribute name="DataType" type="anyURI"/>
</schema>
```

## Дополнение I

## Аспекты безопасности и секретности

Безопасность и секретность должны рассматриваться системно, учитывая человеческие аспекты, такие как социально-технические нападения, аспекты политики, управление ключами и управление доверием, безопасные варианты реализации и другие факторы. Выходящие за рамки настоящего дополнения. Технические решения по безопасности стоят довольно дорого, поэтому должны быть рассмотрены также требования и политические альтернативы, а также законодательные и регуляторные требования.

В настоящем дополнении рассмотрены общие аспекты и подходы безопасности, а также особые угрозы и контрмеры для безопасного, поддерживающего секретность, использования подтверждений SAML, протоколов, связей и профилей. В настоящем дополнении описываются и анализируются свойства SAML по безопасности и секретности. Его назначение состоит в том, чтобы предоставить проектировщикам и создателям систем на базе SAML информацию о следующих вещах:

- вопросах секретности, которые должны быть рассмотрены, и как архитектура SAML учитывает эти проблемы;
- угрозах и, следовательно, рисках для безопасности, для которых система на основе SAML является объектом;
- рисках для безопасности, учитываемых в архитектуре SAML, и как она это делает;
- рисках для безопасности, которые она не учитывает;
- рекомендациях по контрмерам, снижающим эти риски для безопасности.

### I.1 Секретность

Язык SAML имеет возможность делать утверждения об атрибутах и авторизации аутентифицируемых элементов. Существует очень много общих ситуаций, при которых информация, передаваемая в этих утверждениях, представляет собой то, что одна или несколько сторон соединения желали бы оставить доступной для максимально ограниченного круга элементов. Одним из примеров таких ситуация может быть утверждение о медицинском или финансовом атрибуте.

Многие страны и юрисдикции имеют законы и правила относительно секретности, и их следует учитывать при создании системы на базе SAML. Стороны, создающие утверждения, передающие подтверждения, переносящие подтверждения, и использующие подтверждения, должны быть осведомлены о возможных проблемах секретности и должны стараться учитывать их в своих вариантах реализации систем на базе SAML.

### I.2 Конфиденциальность

Возможно наиболее важным аспектом в обеспечении секретности для сторон в транзакции, организованной в среде SAML, является возможность выполнить транзакцию с гарантией конфиденциальности. Другими словами, может ли информация в подтверждении быть передана от создателя до аудитории, которой она предназначена, и только этой аудитории, без возможности доступа к ней каких-либо других сторон?

Технически вполне возможно передать информацию конфиденциально. Все стороны транзакции, организованной в среде SAML, должны анализировать каждый свой шаг в ходе взаимодействия (и все последующие случаи использования данных, полученных в ходе этих транзакций) для обеспечения того, что информация, которая должна оставаться конфиденциальной, действительно таковой остается.

Следует отметить также, что простое скрытие содержания подтверждений может не быть достаточной защитой секретности. Существует множество случаев, когда просто возможность получить информацию о том, что данный пользователь (или IP-адрес) имел доступ к данной услуге, может стать "брешью" в защите секретности (например, информация о том, что данный пользователь обращался в медицинскую лабораторию за подтверждением, может оказаться достаточной для того, чтобы нарушить секретность, даже не зная содержания подтверждений). Частичное решение этих проблем может быть получено с использованием различных методов анонимного взаимодействия, описанных в последующих разделах.

### I.3 Использование псевдонимов и анонимность

Не существует определения анонимности, которое было бы удовлетворительным для всех случаев. Многие определения касаются простого случая с отправителем и сообщением, и рассматривают "анонимность" в смысле "не иметь возможности связать между собой данного отправителя и переданное сообщение или сообщение, переданное обратно отправителю". Несмотря на то что такое определение вполне адекватно для случая "одноразовой" передачи, оно не учитывает возможности накопления информации, которое возможно выполнить на протяжении определенного промежутка времени, на основе выполняемых функций, а не на основе идентификатора.

В языке SAML полезно рассматривать анонимность, как что-то, находящееся "внутри множества". Это обозначение приемлемо для языка SAML, потому что в нем используются ответственные органы. Даже если объект является "анонимным", этот объект все еще может быть идентифицирован как член множества объектов внутри домена определенного ответственного органа. Системы, работающие в среде SAML, могут – в лучшем случае – обеспечить только "частичную анонимность" потому что в нем используются ответственные органы. Элемент, относительно которого создается подтверждение, уже может быть идентифицирован, как один из множества элементов, взаимодействующих с данным ответственным органом.

Ограничения возможностей обеспечения анонимности могут быть гораздо хуже, чем просто связь с ответственным органом, в зависимости от того, как используются идентификаторы и от того, как повторное использование псевдонимов идентификаторов позволяет распространять потенциально идентифицирующей информации. Кроме того, пользователи систем, работающих в среде SAML, могут еще больше нарушать анонимности своими действиями.

Помимо законной функции идентификатора, любой идентификатор объекта можно рассматривать как псевдоним. И даже обозначение типа "держатель ключа" можно рассматривать как эквивалент псевдонима в объединении действия (или множества действий) с Объектом. Даже такое описание как "пользователь, который запросил доступ к объекту XYZ в 23:34" можно рассматривать как эквивалент псевдонима.

Таким образом, учитывая "возможность нанесения вреда", нет никакой разницы между тем, описывается ли пользователь при помощи идентификатора или своим поведением (например, использованием ключа или выполнением действия).

Что их различает, так это то, как часто используется определенный эквивалент псевдонима. Анонимность определяет классификацию псевдонимов, начиная от персональных псевдонимов (типа Ник-имен в сети), которые используются постоянно, до различного вида ролевых псевдонимов (например, секретарь по безопасности), и до псевдонимов "однократного применения".

Только псевдонимы однократного применения могут обеспечить вашу анонимность (в языке SAML учитывайте, что это остается "анонимностью в некотором множестве"). Однако чем чаще используется данный псевдоним, тем больше риска для анонимности. Другими словами, повторное использование псевдонима дает дополнительные возможности для того, чтобы потенциально идентифицирующая информация была бы связана с этим псевдонимом. С течением времени это приведет к увеличению объема информации, что позволит уникальным образом определить идентификатор, связанный с этим псевдонимом.

Ответственные органы на стороне создателя информации (такие как орган по аутентификации и орган атрибута) могут обеспечить определенную степень "частичной анонимности", используя идентификаторы однократного применения или ключи (для случая "держатель ключа"). Такая анонимность, в лучшем случае, является частичной, потому что Объект обязательно привязан к множеству объектов, взаимодействующих с данным ответственным органом. Это множество может быть в дальнейшем сокращено (что еще больше снижает анонимность) за счет объединения используемых атрибутов и это формирует подмножество пользователей на стороне источника. Пользователи, которых интересует сохранение анонимности, должны позаботиться о маскировке и избегать нестандартных действий, которые со времени могут привести к опознаванию их.

## I.4 Безопасность

В последующих разделах рассматриваются аспекты безопасности.

### I.4.1 Теоретические основы

Связь между компьютерными системами является объектом для множества угроз, и эти угрозы несут с собой определенный уровень риска. Природа этого риска зависит от массы факторов, включая вид связи, тип систем связи, тип передачи, физическую среду передачи данных, типы оконечных систем и т. д.

Язык SAML предназначен для того, чтобы помочь создателям систем обеспечить безопасность для прикладного уровня связи между компьютерными системами в рамках данных доменов безопасности или между доменами безопасности. Играя эту роль, SAML передает данные аутентификации, поддерживая возможность оконечных систем защищаться от несанкционированного использования. Безопасность связи непосредственно применяется при разработке применений SAML. Безопасность систем имеет интерес, главным образом, в контексте моделей угрозы языка SAML.

### I.4.2 Сфера применения

Некоторые области, которые заметно влияют на общую безопасность системы, которая использует SAML, находятся за рамками языка SAML. Хотя настоящая Рекомендация не рассматривает эти области, их всегда следует учитывать при рассмотрении проблем безопасности системы. В частности, следующие аспекты очень важны, но в настоящее время находятся за рамками языка SAML.

- Первоначальная аутентификация: SAML позволяет формировать утверждения о действиях по аутентификации, которые выполняются, но не содержит требований или спецификаций для этих действий по аутентификации. Потребители подтверждений аутентификации не должны слепо доверять этим подтверждениям, если только они не знают и до тех пор, пока они не узнают, на базе чего были сделаны эти подтверждения. Доверие к подтверждениям никогда не должно быть выше, чем уверенность в том, что доверяющая сторона корректно получила подтвержденные выводы.
- Модель доверия: Во многих случаях безопасность обмена данными в языке SAML будет зависеть от лежащей в основе модели доверия, которая обычно основана на инфраструктуре управления ключами (например, PKI или секретный ключ). Например, сообщения SOAP, безопасность которых обеспечивается посредством Подписи XML, защищены ровно настолько, насколько можно доверять ключам, используемым при передаче сообщений. Необнаруженные потерявшие секретность ключи или отмененные сертификаты, например, могут представлять собой угрозу безопасности. Даже неудача в получении сертификата может открыть дверь для действий нарушителя под видом законного пользователя. Установка PKI – не тривиальная задача, и она должна выполняться корректно для того, чтобы обеспечить безопасность уровней, построенных на ее основе (например, некоторых частей SAML).

Приемлемые варианты протоколов безопасности должны поддерживать безопасность системы, включая безопасную случайную или псевдослучайную генерацию чисел и безопасное хранение ключей.

### I.4.3 Модель угрозы SAML

Общая модель угрозы интернета, описанная в руководстве IETF по вопросам безопасности, является основой модели угрозы SAML. Мы предполагаем здесь, что две или более оконечные точки транзакции SAML работают без нарушения функционирования системы безопасности, но атакующая сторона имеет полный контроль над каналом связи.

Кроме того, из-за природы SAML, являющегося протоколом многосторонней аутентификация и авторизации, необходимо рассмотреть случаи, при которых одна или несколько стороны в законной транзакции SAML – т. е. те, что вполне законно выполняют свои роли в данной транзакции – пытаются использовать в последующей транзакции информацию, незаконно полученную в ходе предыдущих транзакций.

Следующие ниже сценарии описывают возможные атаки.

- **Сговор**: Секретное взаимодействие между двумя и более элементами системы для выполнения атаки, например:
  - сговор между клиентом и провайдером услуг;
  - сговор между клиентом и провайдером идентификации;
  - сговор между провайдером идентификации и провайдером услуг;
  - сговор между двумя и более клиентами;
  - сговор между двумя и более провайдерами услуг;
  - сговор между двумя и более провайдерами идентификации.
- **Атака "отказ в обслуживании"** (DoS): Недопущение авторизованного доступа к ресурсам системы или задержка действий и выполнения функций системы.
- **Атака типа "человек в середине"**: Форма активной атаки перехвата информации, в которой атакующая сторона перехватывает и выборочно изменяет переданные данные, маскируясь под один или несколько элементов, участвующих в передаче.
- **Атака путем замещения оригинала**: Атака, в которой достоверная передача данных злонамеренно или мошеннически повторяется, либо источником информации, либо злоумышленником, который перехватывает данные и повторно передает их, возможно как часть атаки нападение с маскировкой под законного пользователя.
- **Перехват сеанса связи**: Форма активного перехвата информации, в которой атакующая сторона захватывает контроль над ранее установленным соединением.

Во всех случаях местные механизмы, которые будут использовать системы для принятия решения о том, создавать ли подтверждения, или нет, в настоящей Рекомендации не рассматриваются. Таким образом, угрозы, возникающие из-за подробностей исходного обращения к органу по аутентификации, например, здесь также не рассматриваются. Если ответственный орган передает фальшивое подтверждение, то угрозы, обусловленные использование этого подтверждения нижележащими системами, здесь также не рассматриваются.

Прямым следствием такого ограничения сферы применения является то, что безопасность системы, использующей эти подтверждения в качестве входных данных, хороша ровно настолько, насколько хороша безопасность системы, используемой для создания этих подтверждений, и от корректности данных о процессах их обработки, на которых основаны создаваемые подтверждения. При определении того, каким источникам доверять, особенно в случаях, когда эти подтверждения будут использованы как входные данные для аутентификации или принятия решений об авторизации, риск нарушения безопасности из-за использования фальшивых, но полученных их достоверных источников, подтверждений является чрезвычайно большим. Правила доверия между подтверждающей и доверяющей сторонами всегда должны быть прописаны так, чтобы они содержали аспекты взаимных обязанностей, а их реализация должна содержать соответствующий контрольный журнал.

## I.5 Методы обеспечения безопасности

В последующих разделах описываются методы обеспечения безопасности и различные имеющиеся в наличии технологии, доступные для реализации в системах на базе SAML.

### I.5.1 Аутентификация

Аутентификация в данном контексте означает способность стороны транзакции определить идентификатор другой стороны этой транзакции. Такая аутентификация может быть односторонней или двусторонней.

- **Активный сеанс связи**: Временная аутентификация предоставляется каналом связи, используемым для транспортировки сообщения SAML. Такая аутентификация может быть односторонней – от инициатора сеанса связи к приемнику – или двусторонней. Конкретный метод будет определен используемым протоколом передачи. Например, использование протокола безопасной связи, такого как TLS или IP-security, обеспечивает отправителю сообщения SAML возможность аутентифицировать получателя в среде TCP/IP.
- **Уровень сообщения**: Правила подписи XML Консорциума W3C и документ OASIS WSS описывают методы создания постоянной "аутентификации", которая тесно связана с документом. Этот метод не дает независимой гарантии того, что отправитель сообщения действительно является тем, кто его подписал (и во многих случаях это действительно не так, когда используются промежуточные элементы). Любой метод, который позволяет иметь постоянное подтверждение

участия уникально обозначаемого элемента в данном подмножестве сообщения XML, является достаточным для того, чтобы это требование выполнялось.

### I.5.2 Конфиденциальность

Конфиденциальность означает, что содержание сообщения может быть прочитано только теми, кому оно предназначалось и не сможет быть прочитано никем, кому это сообщение попадется.

- **При передаче**: Использование протокола безопасной связи, такого как TLS или IP-security, обеспечивает временную конфиденциальность сообщения, когда оно передается между двумя узлами.
- **Уровень сообщения**: Шифрование XML обусловливает селективное шифрование документов XML. Этот метод шифрования обеспечивает постоянную селективную конфиденциальность элементов в сообщении XML.

### I.5.3 Целостность данных

Целостность данных это способность подтверждения, что данное сообщение, в том виде, в котором оно получено, не отличается от того варианта сообщения, который был передан.

- **При передаче**: Использование протокола безопасной связи, такого как TLS или IP-security может быть сконфигурировано так, чтобы обеспечить защиту целостности пакетов, передаваемых по сетевому соединению.
- **Уровень сообщения**: Подпись XML обеспечивает метод создания постоянной гарантии неизменности сообщения, который тесно связан с этим сообщением. Любой метод, который позволяет иметь постоянное подтверждение неизменности подмножества данного сообщения XML, является достаточным для того, чтобы это требование выполнялось.

### I.5.4 Замечания об управлении ключами

Многие моменты в настоящем Дополнении будут касаться способности систем обеспечивать аутентификацию, целостность данных и конфиденциальность при помощи различных схем, включая цифровую подпись и шифрование. Для всех этих схем, безопасность, обеспечиваемая схемой, ограничена возможностями существующих систем управления ключами. Ниже подробно рассмотрены некоторые конкретные ограничения.

1) **Доступ к ключу**: Предполагается, что, если системы на основе ключей будут использоваться для обеспечения аутентификации, целостности данных и неотрекаемости от них, то требуются методы безопасности для гарантии того, что доступ к частному или секретному ключу клиента не может быть получен несоответствующими сторонами. Например, цифровая подпись созданная с использованием секретного ключа Боба, является только доказательством того, что Боб участвует в передаче, поскольку только Боб имеет доступ к этому ключу. Как правило, доступ к ключам должен быть у минимально возможного числа элементов (это особенно важно для корпоративных ключей или ключей организации) и он должен быть защищен паролями или иными средствами. Применяются стандартные меры безопасности (не записывайте пароли, если вы уходите от компьютера не оставляйте открытым окно, к которому требуется доступ с ключом и т. д.).

2) **Связь между ключом и идентификатором**: Для систем на основе ключей, которые должны использоваться для аутентификации, должна существовать некоторая доверенная связь между ключом и идентификатором. Проверка цифровой подписи на документе может определить, остался ли документ неизменным после того, как он был подписан с данным ключом. Однако она не подтвердит, что использованный ключ действительно является правильным ключом конкретного человека для этого времени и этой цели. Проверка связи между ключом и стороной требует дополнительных подтверждений.

   Эта связь между ключом и его владельцем должна быть установлена. Широко используемые решения включают в себя локальные справочники, которые содержат и идентификаторы, и ключи – это просто понять, но сложно поддерживать – или использование сертификатов. Использование сертификатов может стать гибким средством для связи ключа с идентификатором, но оно требует механизмов для управления сроком действия сертификата и изменения состояния связи (например, работник увольняется и более не может обладать корпоративным идентификатором). Одним общим подходом является использование инфраструктуры открытых ключей (PKI).

   В этом случае для каждого пользователя подписей определяется множество доверенных корневых органов по сертификации (CA) – они отвечают на вопрос "Кому я доверяю создавать утверждения о связи между ключом и идентификатором?" Теперь проверка подписи превращается сначала в процесс проверки самой подписи (для определения того, что подпись сделана с рассматриваемым ключом, и что сообщение не было изменено) и затем в поверку цепочки создания сертификата (для определения того, что ключ связан с правильной личностью) и проверки того, что эта связь все еще действует. Проверка подписи требует выполнения определенных этапов, гарантирующих, что связь все еще действует – сертификат обычно имеет "срок действия", но если ключ раскрыт во время срока действия, то связь между ключом и идентификатором, указанная в сертификате, становится недействительной, хотя сам по себе сертификат остается действительным. Кроме того, сертификаты часто зависят от связей, которые могли прекратить существование до того, как закончился срок действия сертификата (например, сертификаты, которые должны стать недействительными, когда меняются работники и т. д.). Таким образом, правильная система управления ключами достаточно сильна, но и очень сложна. Проверка подписи оказывается процессом проверки связи между документом и ключом, затем проверки связи между ключом и идентификатором, а также проверки действительности ключа и сертификата.

### I.5.5 Модули шифрования TLS

Во многих частях настоящей Рекомендации настоятельно рекомендуется использовать HTTP поверх SSL 3.0 (см. Дополнение IV) или TLS 1.0, или использование идентификаторов URL со схемой HTTPS URL.

Если в спецификации не определено иного, то при использовании любой связи SAML, серверы SSL (Secure Sockets Layer – Протокол для создания криптографически защищенных сетевых соединений) 3.0 или TLS 1.0 (Transport Layer Security – Безопасность на транспортном уровне) должны аутентифицировать себя для клиента, используя сертификат X.509 v3. Клиент должен установить идентификатор сервера на основе содержания сертификата (обычно путем проверки поля DN в объекте сертификата).

Протоколы SSL/TLS могут быть сконфигурированы так, чтобы использовать различные модули шифрования, не все из которых пригодны для обеспечения "лучшей из опыта" безопасности. Модуль шифрования объединяет четыре вида возможностей обеспечения безопасности и имеет имя в [SSL]. До передачи потоков данных по соединению SSL, обе стороны пытаются согласовать модуль шифрования. Это позволяет им установить требуемое качество защиты для их связи с ограничениями, свойственными комбинациям конкретного доступного механизма. Возможности, связанные с модулем шифрования таковы:

Протокол SSL определяет множество алгоритмов обмена ключами. Некоторые механизмы выполняют аутентификацию сервера. Однако поддерживаются также и механизмы анонимного обмена ключами. (Алгоритмы анонимного обмена ключами являются объектом атак "человек в середине" и не рекомендуются к использованию в среде SAML.) Алгоритм обмена ключами, аутентифицированный при помощи алгоритма "RSA" в настоящее время является наиболее широко используемым алгоритмом (истек срок действия патента на алгоритм RSA). Другим важным алгоритмом обмена ключами является алгоритм, аутентифицированный при помощи алгоритма Диффи-Хеллмана "DHE_DSS", для которого нет ограничений на использование, связанных с патентами.

Любой из этих алгоритмов обмена ключами может быть свободно экспортирован из США. Экспортируемые алгоритмы должны использовать короткие (512-битовые) открытые ключи для обмена ключами и короткие (40-битовые) симметричные ключи для шифрования. Ключи такой длины были успешно атакованы и их использование не рекомендуется.

Наиболее быстрым алгоритмом шифрования является потоковый шифр RC4; DES и его варианты (DES40, 3DES-EDE), а также AES также поддерживаются в режиме "блочного шифра со сцеплением блоков" (CBC). Другие режимы также поддерживаются в соответствии с документацией TLS.

Нулевое шифрование является опцией в некоторых модулях шифрования. Нулевое шифрование не предусматривает никакого шифрования; в таких ситуациях протокол SSL/TLS используется только для аутентификации и защиты целостности. Модули шифрования с нулевым шифрованием не обеспечивают конфиденциальности и не должно использоваться в тех случаях, когда требуется конфиденциальность, которая не обеспечивается средствами, отличными от SSL/TLS.

Краткий алгоритм, используемый для кода сообщения аутентификации. Федеральная комиссия связи США (FCC) недавно рекомендовала применять алгоритм SHA-256, и рабочая группа по стандартам для сети Internet (IETF) решила следовать этой рекомендации.

## I.6 Общие аспекты безопасности языка SAML

В последующих разделах анализируются риски для безопасности при использовании и реализации SAML и описываются контрмеры для снижения рисков.

### I.6.1 Подтверждения SAML

На уровне самого подтверждения SAML, мало, что можно сказать по вопросу безопасности – большая часть проблем возникает во время связи по протоколу запрос/ответ, или во время попытки использовать SAML в одной из связей. От потребителя, конечно, всегда ожидается учет срока действия подтверждения и любых элементов `<OneTimeUse>`, которые представлены в подтверждении.

Однако один из аспектов на уровне подтверждение требует анализа: подтверждение, после того как оно создано, выходит из-под контроля своего создателя. Этот факт имеет массу последствий. Например, создатель не может контролировать, как долго это подтверждение будет существовать в системе потребителя; точно также создатель не может контролировать, с какими сторонами потребитель будет совместно использовать информацию из этого подтверждения. К этим проблемам добавляются проблемы злоумышленника, который может видеть содержание подтверждений, передаваемых в канале незашифрованными (или недостаточно зашифрованными).

Хотя были предприняты усилия по разрешению многих из этих проблем в рамках Рекомендации SAML, ничто из того, что имеется в настоящей Рекомендации, не может устранить требований по тщательному учету того, что должно быть помещено в подтверждение. В любой момент создатели подтверждений должны учитывать возможные последствия того, что информация, представленная в подтверждении, может храниться на удаленном сайте, где она может быть использована злоумышленниками или оказаться доступной для потенциальных хакеров, или, возможно, сохранена для более изощренного мошеннического использования. Создатели подтверждений должны также учитывать возможные последствия того, что информация в подтверждении может быть использована совместно с другими сторонами или даже стать общедоступной, как преднамеренно, так и по недосмотру.

### I.6.2 Протокол SAML

В настоящем разделе описываются аспекты безопасности для самого протокола SAML запрос-ответ, за исключением любых угроз, обусловленных использованием определенного протокола связи.

– **Отказ в обслуживании**

Протокол SAML чувствителен к атаке типа "отказ в обслуживании" (DoS). Обработка запроса SAML – это очень дорогостоящая операция, включающая синтаксический разбор сообщения-запроса (обычно предусматривающий создание дерева Объектной модели документа (DOM)), просмотр базы данных подтверждений (возможно неиндексированной), создание сообщения-ответа и, возможно, одна или несколько операций с цифровой подписью. Таким образом, усилие, требуемое от атакующего для создания запроса, намного меньше усилий по обработке этих запросов.

**1) Требуемая аутентификация клиента на низком уровне**

Требуемая аутентификация клиента на некотором уровне ниже уровня протокола SAML (например, с использованием связи SOAP поверх HTTP, с HTTP поверх TLS/SSL и с требованием наличия сертификатов на стороне клиента, которые заслуживают доверия со стороны ответственного органа по сертификации) обеспечит возможность отслеживания нападения в случае атаки DoS.

Если аутентификация используется только для обеспечения возможности отслеживания, то она не сможет предотвратить атаку, но будет действовать, как средство устрашения.

Если аутентификация объединена с какой-либо системой контроля доступа, тогда атаки DoS со стороны внешних пользователей эффективно блокируются. (Возможно, что перегрузка схемы аутентификации клиента будет продолжать действовать как атака "отказ в обслуживании" для службы SAML, но эта атака должна быть отражена выбранной схемой аутентификации клиента.)

Какая бы ни использовалась система аутентификации, она должна предоставить возможность опознать уникального создателя каждого запроса и не должна быть объектом фальсификации. (Например, в случае использования только для обеспечения возможности отслеживания, загрузка IP-адреса е является достаточной, поскольку эту информацию легко подделать.)

**2) Требуемые подписанные запросы**

Требуемый подписанный запрос также снижает степень асимметрии между работой, выполняемой запрашивающей и отвечающей сторонами. Дополнительная работа, которую должна выполнить отвечающей сторона для проверки подписи, составляет относительно малый процент от общей работы, которую должна выполнить отвечающая сторона, тогда как процесс расчета цифровой подписи представляет собой относительно большой процент работы, которую должна выполнить запрашивающая сторона. Снижение этой асимметрии уменьшает риск атаки DoS.

Однако поскольку атакующая сторона, теоретически, может перехватить подписанное сообщение и затем повторять его непрерывно в попытках обойти это требование. Этой ситуации можно избежать, если потребовать использовать элемент подписи XML `<ds:SignatureProperties>`, содержащий метку времени; эта метка времени может быть затем использована для определения того, не устарела ли подпись. В этом случае чем меньше промежуток времени после создания этой подписи, в течение которого подпись остается действительной, тем выше безопасность в отношении атаки путем замещения оригинала и атаки отказа в обслуживании.

**3) Ограниченный доступ к URL взаимодействия**

Ограничение возможности создания запроса на услугу SAML на очень низком уровне до набора известных сторон существенно снижает риск атаки DoS. В этом случае возможны только атаки, созданные в ограниченном наборе известных сторон, что существенно снижает как опасность со стороны клиентов-злоумышленников, так и опасность атак DoS, использующих взломанные компьютеры без их на то ведома.

Существует множество возможных методов ограничения доступа, например, помещение отвечающей стороны SAML внутрь защищенной сети интранет и реализация правил доступа на уровне маршрутизатора.

## I.7 Аспекты безопасности связей языка SAML

Аспекты безопасности при разработке протокола SAML запрос-ответ во многом зависят от конкретной связи протокола, которая используется. Поддерживаются следующие связи: связь SOAP, связь Reverse SOAP (PAOS), связь HTTP Redirect, связь HTTP Redirect/POST, связь HTTP Artifact и связь SAML URI.

### I.7.1 Связь SAML SOAP

Поскольку связь SAML SOAP не требует аутентификации, и не имеет требований по конфиденциальности в процессе передачи, ни по целостности сообщения, она открыта для широкого многообразия обычных атак. Общие аспекты рассматриваются отдельно от аспектов для случая SOAP-поверх-HTTP.

**1) Перехват информации**

**Угроза**: Поскольку здесь нет требований по конфиденциальности в процессе передачи, вполне возможно, что сторона, перехватывающая информацию, сможет получить как сообщение SOAP, содержащее запрос, так и сообщение SOAP содержащее соответствующий ответ. Такой перехват раскрывает как вид запроса, так и подробности ответа, возможно также несколько подтверждений.

Раскрытие подробностей запроса в некоторых случаях ослабит безопасность запрашивающей стороны, раскрывая детали того, какие типы подтверждений она запрашивает, или от кого их требует. Например, если станция перехвата может определить, что сайт *X* часто запрашивает

подтверждения аутентификации с определенным методом подтверждения от сайта *Y*, она может использовать эту информацию для взлома сайта *X*.

Аналогично, перехват информации о серии запросов авторизации может создать "карту" ресурсов, находящихся в компетенции данного ответственного органа по авторизации.

Кроме того, в некоторых случаях раскрытие самого запроса может быть нарушением секретности. Например, перехват информации о запросе и ответе на него может раскрыть сведения о том, что данный пользователь действует на запрашивающем сайте, что может оказаться той информацией, которая не должна разглашаться, например, если это касается медицинских информационных сайтов, политических сайтов и т. д. Кроме того, подробности любого подтверждения, содержащиеся в ответе, могут быть той информацией, которая должна храниться в секрете. Это особенно важно для ответов, содержащих подтверждения атрибутов; если эти атрибуты представляют собой данные, которые не должны быть доступны для сторон, не участвующих в транзакции (кредитные ставки, медицинские данные и т. д.), тогда риск, обусловленный перехватом информации, очень высок.

**Контрмеры**: В тех случаях, когда возможен любой из этих рисков, контрмеры, противостоящие атакам перехвата информации должны обеспечить какую-либо форуму конфиденциальности сообщений в процессе передачи. Для сообщений SOAP такая конфиденциальность может быть обеспечена либо на уровне SOAP, либо на транспортном уровне SOAP (или на уровне ниже него).

Добавление конфиденциальности в процессе передачи на уровне SOAP означает создание сообщения SOAP таким образом, чтобы, вне зависимости от транспорта SOAP, никто кроме адресата не мог бы получить доступ к сообщению. Общим решением этой проблемы, вероятно, будет шифрование XML. Настоящая Рекомендация допускает шифрование самого сообщения SOAP, что устраняет риск перехвата информации, если только не был раскрыт ключ, используемый для шифрования. В качестве альтернативы, для обеспечения конфиденциальности в процессе передачи проектировщики могут использовать транспортный уровень SOAP или уровень под ним.

Подробные данные о том, как обеспечить такую конфиденциальность, зависят от конкретного выбранного транспорта SOAP. Одним из методов валяется использование HTTP поверх TLS/SSL. Другие виды транспорта обусловят необходимость других методов обеспечения конфиденциальности в процессе передачи; например, транспорт SMTP может использовать S/MIME.

В некоторых случаях, уровень ниже транспортного уровня SOAP может обеспечить требуемую конфиденциальность в процессе передачи. Например, если взаимодействие запрос-ответ выполняется в туннеле IPSec, то адекватная конфиденциальность в процессе передачи может быть обеспечена самим туннелем.

**2) Повторная передача**

**Угроза**: На уровне связи SOAP опасность атаки повторной передачи невелика. Повторная передача представляет большую угрозу в различных профилях. Основное беспокойство относительно повторной передачи на уровне связи SOAP представляет собой возможность использования повторной передачи как метода атаки "отказ в обслуживании".

**Контрмеры**: Как правило, наилучшим способом предотвратить атаку повторной передачи является предотвращение возможности перехвата сообщения. Некоторые схемы транспортного уровня, используемые для обеспечения конфиденциальности в процессе передачи, обеспечат достижение этой цели. Например, если связь SAML в режиме запрос-ответ осуществляется в среде SOAP на HTTP/TLS, третьи стороны не имеют возможности перехвата сообщения.

Поскольку потенциальному злоумышленнику не требуется понимать сообщение для того, чтобы выполнить его повторную передачу, схемы типа шифрования XML не обеспечивают защиты от повторной передачи. Если атакующая сторона может перехватить запрос SAML. Который был подписан запрашивающей стороной и зашифрован для отвечающей стороны, то атакующая сторона сожжет передать этот запрос повторно в любой момент без его дешифровки. Запрос SAML содержит информацию о времени создания запроса, которая позволяет выявить наличие повторной передачи. Кроме того, для выявления повторной передачи запроса может использоваться уникальный ключ запроса (его `ID`).

Дополнительные угрозы от атаки повторной передачи включают в себя случаи, при которых используется модель "оплата по числу запросов". Повторная передача может использоваться для увеличения счетов, выставляемых по данному аккаунту.

Аналогично, модели, в которых клиенту назначается (или клиент покупает) фиксированное число взаимодействий с системой, атаки повторной передачи могут использовать все возможности, если только создатель запросов не ведет тщательного учета уникальных ключей для каждого запроса.

**3) Введение сообщения**

**Угроза**: Сфабрикованный запрос или ответ вводится в поток сообщений. Фальшивый ответ, например поддельная повторная передача ответа "Да" на запрос с неавторизованным решением или возврат фальшивой информации об атрибуте в ответ на запрос атрибута могут привести к неправильным действиям приемника.

**Контрмеры**: Способность ввести запрос – это не угроза на уровне связи SOAP. Угрозой введения фальшивого ответа может быть атака отказ в обслуживании, например возврат в качестве ответов сообщения об ошибке SOAP, но такая атака может быстро стать очевидной. Более изощренная атака, предусматривающая возврат сфабрикованных ответов, рассматривается в протоколе SAML, она вполне приемлема, поскольку в соответствии с определением связи SOAP, каждый ответ SOAP должен содержать один-единственный ответ по протоколу SAML, если он не содержит ошибки.

Протокол SAML решает эти проблемы при помощи двух механизмов, установления соответствия ответов запросам с использованием требуемого атрибута `InResponseTo`, что усложняет атаку, поскольку для того, чтобы создать ответ, нужно перехватить еще и запрос, и поддержки аутентификации источника, либо при помощи подписанных ответов SAML, либо при помощи безопасного транспортного соединения, такого как SSL/TLS.

**4) Удаление сообщения**

**Угроза**: Атака удаления сообщения либо не позволит запросу достичь отвечающей стороны, либо не позволит ответ достичь запрашивающей стороны.

**Контрмеры**: В любом случае, связь SOAP не противостоит этой угрозе. Как правило, корреляция между сообщениями запрос и ответ может остановить такую атаку, например использование атрибута `InResponseTo` в элементе **StatusResponseType**.

**5) Изменение сообщения**

**Угроза**: Изменение сообщения – это угроза для связи SOAP в обоих направлениях.

Изменение запроса с изменением деталей запроса может привести к тому, что будут возвращены существенно иные результаты, что, в свою очередь, может использоваться умным злоумышленником для взлома систем в зависимости от возвращенных подтверждений. Например, изменение списка запрошенных атрибутов в элементах `<Attribute>` может привести к получению результатов, приводящих к искажению запроса или получении отказа на него от отвечающей стороны.

Изменение запроса с изменением данных о создателе запроса может привести к отказу в обслуживании или неправильной маршрутизации ответа. Такое изменение может проявиться на уровне ниже уровня SAML и, таким образом, здесь не рассматривается.

Изменение ответа с изменением деталей подтверждения может привести к высокой степени взлома. Простые примеры искажения деталей аутентификации или решения об авторизации может привести к очень серьезным угрозам безопасности.

**Контрмеры**: Для того чтобы противостоять этим возможным угрозам, должна использоваться система, которая гарантирует целостность сообщения в процессе передачи. Ни протокол SAML, ни связь SOAP не требуют и не запрещают использование систем, гарантирующих целостность сообщения в процессе передачи, то в связи с размерами этой угрозы, настоятельно рекомендуется использовать такую систему. На уровне связи SOAP это может быть выполнено при помощи цифровой подписи запросов и ответов, например Подписью XML.

Если сообщения подписаны цифровой подписью, то получатель имеет гарантию того, что это сообщение не было изменено в процессе передачи, если только не был взломан используемый ключ.

Цель обеспечения целостности сообщения в процессе передачи может быть достигнута на более низком уровне, если используется транспорт SOAP, который имеет возможность обеспечения целостности, и связь основана на протоколе, который обеспечивает такую целостность. Транспортом, который даст такую гарантию, является SOAP поверх HTTP поверх TLS/SSL.

Шифрование само по себе такой защиты не обеспечивает, даже если перехваченное сообщение не может быть изменено *само по себе*, оно может быть заменено другим – вновь созданным.

**6) Человек в середине**

**Угроза**: Связь SOAP подвержена атаке "человек в середине" (MITM). Для того чтобы не дать возможности злоумышленникам действовать по схеме "человек в середине" (которая имеет все опасности, которые были рассмотрены в разделах, касающихся перехвата информации и изменения сообщения), требуется взаимная аутентификация какого-либо вида.

**Контрмеры**: Система двусторонней аутентификации позволит обеим сторонам определить, что то, что они видят в сеансе связи, действительно получено о другого участника сеанса связи.

На уровне связи SOAP, эта цель может быть достигнута за счет цифровой подписи и запросов, и ответов. Этот метод не устранит возможности того, что станция перехвата расположена между передатчиком и приемником, и передает сообщения в обе стороны, но он устранит возможность изменения текста переговоров без обнаружения такого изменения.

Поскольку во многих приложениях протокола SOAP сеансы связи не используются, этот вид аутентификации автора (в противоположность аутентификации отправителя) может потребоваться скомбинировать с информацией от транспортного уровня для подтверждения того, что автор и отправитель это она и та же сторона, для того чтобы предотвратить возможность реализации более слабой версии " Человек в середине как станция перехвата".

Другой вариант реализации будет зависеть от транспорта SOAP, с помощью которой он осуществлен или реализован на более низком уровне, который обеспечивает двустороннюю аутентификацию. Примером этого может снова служить SOAP поверх HTTP поверх TLS/SSL, где требуются сертификаты и на стороне клиента, и на стороне сервера.

Кроме того, интервал достоверности функций возвращенного подтверждения является дополнительным изменением степени риска атаки MITM. Чем короче интервал достоверности, тем меньше вреда будет нанесено при перехвате.

**7) Использование SOAP поверх HTTP**

Поскольку связь SOAP требует, чтобы соответствующие ей приложения поддерживали протокол передачи "HTTP поверх TLS/SSL" с множеством различных методов двусторонней аутентификации, таких как Basic поверх SSL на стороне сервера и аутентификация на базе

сертификатов поверх SSL на стороне сервера, эти методы всегда доступны для смягчения угроз в тех случаях, когда другие системы более низкого уровня недоступны, а вышеперечисленные атаки, как предполагается, представляют значительную угрозу.

Это не означает, что использование HTTP поверх TLS с двусторонней аутентификацией некоторого вида является обязательным. Если приемлемый уровень защиты от различных рисков может быть обеспечен при помощи иных средств (например, при помощи туннеля IPSec), полного TLS с сертификатами не требуется. Однако в большинстве случаев для SOAP поверх HTTP, хорошим выбором будет использование HTTP поверх TLS с двусторонней аутентификацией.

Документ RFC об аутентификации HTTP (IETF RFC 2617) описывает возможные атаки в условиях HTTP, когда используются базовые схемы аутентификации или схемы аутентификации на основании профиля сообщения.

Отметим, однако, что использование методов обеспечения безопасности на уровне транспорта (таких как протоколы SSL или TLS в условиях HTTP) обеспечивает только конфиденциальность и/или целостность и/или аутентификацию для одного "скачка". Модели, где могут присутствовать промежуточные элементы или рассматриваемые подтверждения, должны существовать дольше, чем один "скачок", и использование HTTP с TLS/SSL не обеспечивает адекватной безопасности.

### I.7.2 Профили единой регистрации в сети (SSO) для веб-браузера

Аутентификация пользователя на стороне источника выходит за рамки настоящей Рекомендации, как и аспекты, касающиеся аутентификации этого сайта источника. Ключевым является то, что элемент системы источник должен иметь возможность подтвердить, что элемент системы аутентифицированного клиента, с которым он взаимодействует, является тем же самым, что и элемент на следующем этапе взаимодействия. Одним из способов сделать это заключается в том, чтобы выполнить эти начальные этапы с использованием TLS в качестве уровня сеанса связи в основе протокола, применяемого для этого первоначального взаимодействия (вероятно, HTTP).

#### I.7.2.1 Профиль SSO

**1) Перехват информации**

**Угроза**: Возможность перехвата информации существует для всех случаев применения веб-браузера.

**Контрмеры**: В тех случаях, когда требуется конфиденциальность (учитывая, что ни одно подтверждение не передается полностью безопасно, и, вместе с запросами, которые с ним ассоциированы, доступно для станции перехвата злоумышленников), требуется, чтобы трафик HTTP передавался средствами транспортировки, которые обеспечивают конфиденциальность. Этому требованию удовлетворяют HTTP поверх TLS/SSL и протокол IP-security.

**2) Похищение информации аутентификации пользователя**

**Угроза**: В том случае, когда объект аутентифицирует себя для сайта источника, раскрывая информацию аутентификации, которую можно использовать повторно, например, в виде пароля, похищение информации аутентификации даст возможность противнику выдать себя за этот объект.

**Контрмеры**: Для того чтобы избежать этой проблемы, соединение между браузером объекта и сайтом источника должно осуществляться с защитой конфиденциальности. Кроме того, до передачи информации аутентификации либо объектом, либо адресатом должны быть предприняты меры для гарантии того, что сайт источника действительно является ожидаемым и доверенным сайтом источника. Для решения этой проблемы может использоваться HTTP поверх TLS.

**3) Похищение метки канала передачи**

**Угроза**: В том случае, когда подтверждение аутентификации содержит подтверждение идентификатора протокола аутентификации канала передачи, похищение этого артефакта даст возможность противнику выдать себя за этот объект.

**Контрмеры**: Все приведенные ниже методы снижают вероятность того, что случится:

На сайте адресата реализованы меры защиты конфиденциальности при соединении его с браузером объекта.

Сайт адресата или источника гарантирует (дополнительно), что на сайте источника реализованы меры защиты конфиденциальности при соединении его с браузером объекта.

Сайт адресата проверяет, чтобы браузер объекта был бы непосредственно перенаправлен сайтом источника, который сам аутентифицировал объект.

Сайт источника отказывается отвечать, если получает несколько запросов на соответствующее подтверждение с одним и тем же ID подтверждения.

Если подтверждение содержит элемент "условие" типа **AudienceRestrictionType**, который идентифицирует определенный домен, то сайт адресата проверяет, чтобы он был членом этого домена.

Соединение между сайтом адресата и сайтом источника, по которому передается ID подтверждения, реализуется с мерами защиты конфиденциальности.

Сайт адресата, в своем соединении сайтом источника, по которому передается ID подтверждения, должен проверить, что сайт источника действительно является ожидаемым и доверенным сайтом источника.

**4) Повторная передача**

Для этого множества профилей существует вероятность атаки типа Повторная передача. Атака Повторная передача может использоваться либо, как попытка отказа в обслуживании либо для мошеннического получения информации. Конкретные контрмеры зависят от того, какая конкретно связь используется, и они рассмотрены выше.

**5) Введение сообщения**

Атаки типа Введение сообщения рассмотрены в I.7.1.

**6) Удаление сообщения**

**Угроза**: Удаление сообщения на любом этапе взаимодействия между браузером, создателем подтверждений SAML и потребителем подтверждений SAML приведет к тому, что взаимодействие прекратится. Это приводит к задержке в предоставлении услуг, но не повышает вероятности раскрытия какой-либо информации.

**Контрмеры**: Использование канала транспортировки, обеспечивающего целостность сообщения, успешно противостоит угрозе удаления сообщения, когда нет промежуточных элементов.

**7) Изменение сообщения**

**Угроза**: Для этого множества профилей существует вероятность изменения сообщения в передаваемом потоке. Могут быть получены следующие нежелательные результаты:

Изменение исходного запроса может привести к отбрасыванию создателя сообщения SAML, или к созданию артефакта, нацеленного на ресурс, отличный от запрашиваемого ресурса.

Изменение артефакта может привести к отказу в обслуживании пользователя SAML.

Изменение самих подтверждений в процессе передачи может привести к нежелательным результатам всех видов (если они не подписаны) или отказу в обслуживании (если они подписаны, а потребитель отбросил их).

**Контрмеры**: Для того чтобы избежать возможности изменения сообщений, трафик должен транспортироваться при помощи систем, которые гарантируют целостность сообщения от одной оконечной точки до другой оконечной точки.

Для профилей на базе веб-браузеров, рекомендованным методом обеспечения целостности сообщения в процессе передачи является использование HTTP поверх TLS/SSL с модулем шифрования, который обеспечивает проверку целостности данных.

**8) Человек в середине**

**Угроза**: Атаки типа "человек в середине" особенно опасны для данного множества профилей. Атака MITM может ретранслировать запросы, перехватывать возвращаемые подтверждения (или артефакты) и передавать обратно фальшивые сообщения. При этом настоящий пользователь не может получить доступ к требуемому ресурсу, а MITM может сделать это, используя перехваченный ресурс.

**Контрмеры**: Для предотвращения этой угрозы требуется множество контрмер. Во-первых, использование системы, которая обеспечивает сильную двустороннюю аутентификацию, затруднит MITM задачу входа в соединение.

Однако остается вероятность, что MITM, который просто, действуя как двунаправленный порт, ретранслирует и перехватывает информацию о передаваемой информации с целью перехватить возвращаемое подтверждение или обработчик (и, возможно, изменить окончательную версию, возвращаемую запрашивающей стороне). Внедрение системы защиты конфиденциальности предотвратит перехват информации. Внедрение системы защиты целостности данных предотвратит изменение сообщения на этапе его пересылки.

Для этого множества профилей все требования относительно двусторонней аутентификации связи, конфиденциальности и целостности данных могут быть выполнены за счет применения HTTP поверх TLS/SSL, если уровень TLS/SSL использует соответствующий модуль шифрования (достаточно устойчивое шифрование для обеспечения конфиденциальности и поддержки целостности данных) и требуют применения для аутентификации сертификатов X.509 v3.

**9) Подмена без повторной аутентификации**

**Угроза**: Мошенник пытается выдать себя за подключившегося законного клиента и получить таким образом доступ к защищенным ресурсам.

После того как клиент успешно подключился к провайдеру идентификации, следующие сообщения `<AuthnRequest>` от других провайдеров услуг, касающихся этого клиента, не обязательно приведут к повторной аутентификации. Клиент, однако, должен быть аутентифицирован, если только провайдер идентификации не сможет определить, что `<AuthnRequest>` связан не только с идентификатором клиента, но также и с достоверностью сеанса связи аутентифицированного провайдера идентификации для этого клиента.

**Контрмеры**: В реализациях, где эта угроза может быть реальной, провайдеры идентификации должны сохранять информацию о состоянии активных сеансов связи и, прежде чем аутентифицировать клиента, должны проверить всю переписку между `<AuthnRequest>` и активным сеансом связи до передачи сообщения `<Response>`. Маркеры HTTP, переданные провайдером идентификации могут использоваться для поддержки этого проверочного процесса, хотя альянс Liberty не поддерживает подход на базе маркеров HTTP.

#### I.7.2.2 Расширенный профиль клиента и прокси-сервера

**1) Человек в середине**

**Угроза**: Перехват сообщений SOAP `AuthnRequest` и `Response`, позволяющий имитировать последующего клиента.

Вредоносный элемент системы может ввести самого себя в положение "человек в середине" (MITM) между расширенным клиентом и законным провайдером услуг, где он играет роль провайдера услуг, взаимодействуя с расширенным клиентом, и роль расширенного клиента, взаимодействуя с законным провайдером услуг. Таким образом, в качестве первого шага, MITM способен перехватить запрос провайдера услуг `AuthnRequest` и заменить любое значение URL по своему выбору на значение `responseConsumerServiceURL` в блоке заголовка PAOS, прежде чем пересылать `AuthnRequest` расширенному клиенту. Обычно MITM будет вводить значение URL, которое указывает на него. Затем, если расширенный клиент принимает ответ от провайдера идентификации и затем передает содержимое ответа на `responseConsumerServiceURL`, принятый от MITM, то MITM получает возможность выдать себя за клиента для законного провайдера услуг.

**Контрмеры**: Провайдер идентификации определяет для расширенного клиента адрес, по которому расширенный клиент должен передавать свой ответ. Значение `responseConsumerServiceURL` в блоке заголовка PAOS используется только для ошибочных ответов от расширенного клиента – как определено в спецификации профиля.

**2) Отказ в обслуживании**

**Угроза**: Изменение запроса SOAP `AuthnRequest` таким образом, что он не может быть обработан, например, изменение атрибута значения услуги в блоке заголовка PAOS на неизвестное значение или изменение блока заголовка ECP `ProviderID` или `IDPList` приведет к невозможности выполнить запрос.

**Контрмеры**: Обеспечить защиту целостности для сообщения SOAP, используя протокол SOAP Message Security или SSL/TLS.

#### I.7.2.3 Профиль обнаружения провайдера идентификации

**Угроза**: Атака в виде фальшивых маркеров HTTP, когда изменяются параметры маркеров HTTP, для того чтобы был обнаружен мошеннический провайдер идентификации, например.

**Контрмеры**: Специальный механизм использования общего домена ограничит возможности этой угрозы.

#### I.7.2.4 Профиль единого выхода из системы

**Угроза**: Пассивная атакующая сторона может собрать данные об идентификаторе имени клиента.

В ходе начальных шагов, пассивная атакующая сторона может собрать информацию `<LogoutRequest>`, когда она передается. Раскрытие этих данных представляет собой угрозу секретности.

**Контрмеры**: Все передачи должны выполняться по безопасному транспорту, такому как SSL или TLS.

**Угроза**: Неподписанное сообщение `<LogoutRequest>`.

Вредоносный элемент системы может ввести неподписанное сообщение `<LogoutRequest>`, запретив таким образом предоставление услуги клиенту. Если предположить, что `NameID` может быть угадан или выведен, тогда понятно, что агент пользователя может получить указание доставить сфабрикованное сообщение `<LogoutRequest>`.

**Контрмеры**: Подписать сообщение `<LogoutRequest>`. Провайдер идентификации может также проверить идентификатор клиента клиент при отсутствии подписанного запроса.

#### I.7.2.5 Профили управления идентификатором имени

**Угроза**: Разрешение элемента системы кореллировать информацию или, в ином случае, недопустимо раскрывать информацию об идентификации, нарушая требования секретности.

**Контрмеры**: IdP дожжен позаботиться о том, чтобы использовать для одного и того же клиента различные идентификаторы имен с различными провайдерами услуг. IdP должен зашифровать идентификатор имени, который он возвращает провайдеру услуг, с тем, чтобы при последующих взаимодействиях использовался скрытый идентификатор.

#### I.7.2.6 Профили атрибутов

Угрозы, связанные со связями, ассоциированными с профилями атрибутов, уже рассмотрены выше. Не известно никаких дополнительных особых угроз, определяемых этими профилями.

## Дополнение II

## Регистрация типа канала передачи MIME application/samlassertion+xml

В настоящем дополнении содержится регистрация типа канала передачи MIME `application/samlassertion+xml`.

**Название типа канала передачи MIME**

– `application`

**Название подтипа MIME**

– `samlassertion+xml`

**Требуемые параметры**

– Нет.

**Дополнительные параметры**

– `charset`

– Тот же, что и в параметре `charset` со значением `application/xml`, как в IETF RFC 3923.

**Аспекты кодирования**

– Те же, что и для `application/xml`, как в IETF RFC 3923.

**Аспекты безопасности**

Объекты типа `samlassertion+xml` не содержат исполнимого контента. Однако подтверждения SAML являются объектами XML. Раз так, то в них имеются все аспекты безопасности, которые представлены в IETF RFC 3923, раздел 10, а также дополнительные аспекты, поскольку они являются явными объектами безопасности. Например, объекты типа `samlassertion+xml` часто будут содержать данные, которые могут определять или относиться к реальному человеку, и могут использоваться в качестве основы для сеансов связи и решений контроля доступа.

Для того чтобы противостоять возможным проблемам, объекты типа `samlassertion+xml` содержат данные, которые должны быть соответствующим образом подписаны отправителем. Любая такая подпись должна быть заверена получателем данных – и как достоверная подпись, и как подпись отправителя. Создатели объектов типа `samlassertion+xml`, содержащих подтверждения SAML, могут также зашифровать, все подтверждение, или его часть.

Кроме того, профили SAML и связи протокола определяют, при необходимости, использование безопасных каналов.

Язык SAML Версия 2 (настоящая Рекомендация) содержит в своем проекте различные методы обеспечения секретности. Например: объектам могут быть назначены скрытый ссылки, присущие взаимодействиям между определенными элементами системы. Эти ссылки могут трансформироваться в идентификаторы с более широким смыслом (например, адреса электронной почты, идентификаторы аккаунта и т. д.) только определенными сторонами.

**Аспекты взаимодействия**

Подтверждения SAML имеют явно указанную версию. Доверяющие стороны должны гарантировать, что они учитывают информацию о версии подтверждения и ведут себя соответственно.

**Опубликованная спецификация**

Язык SAML Версия 2 (настоящая Рекомендация) явно определяет использование типа канала передачи MIME `application/samlassertion+xml`. Однако совершенно понятно, что подтверждения не-SAML (т. е. SAMLv1 и/или SAMLv1.1) на практике могут передаваться с использованием связей языка SAML.

**Приложения, которые используют этот тип канала передачи**

Потенциально, любое приложение, в котором реализован SAML, а также те приложения, в которых используются спецификации, основанные на SAML.

**Дополнительная информация**

**Магический(е) номер(а)**

Как правило, они те же самые, что и для `application/xml`. В частности, корневой элемент XML возвращенного объекта будет иметь имя, определенное в области имен с:

- локальным именем: `Assertion`;

- область имен URI: одна из областей имен URI, определенная версией SAML подтверждения XML, как определено в соответствующей "базовой" Рекомендации SAML для этой версии.

Особенно для SAML, корневой элемент возвращенного объекта может быть либо `<saml:Assertion>`, либо `<saml:EncryptedAssertion>`, где "saml" представляет собой любой префикс области имен XML, который преобразуется в область имен URI подтверждения SAML:

`urn:oasis:names:tc:SAML:2.0:assertion`

**Расширение(я) файла**

Нет.

**Код(ы) файлов Макинтош**

Нет.

**Человек и его электронный адрес для контактов для получения информации**

Данная регистрация выполнена от имени Технического комитета OASIS по Службам безопасности (SSTC).

**Предназначение**

Общего пользования.

## Дополнение III

## Регистрация типа канала передачи MIME application/samlmetadata+xml

В настоящем дополнении содержится регистрация типа канала передачи MIME – `application/samlmetadata+xml` – для использования с сериализацией языка разметки XML, предусматривающего защиту данных метаданных.

**1) Название типа канала передачи MIME**

– `application`

**2) Название подтипа MIME**

– `samlmetadata+xml`

**3) Требуемые параметры**

– Нет.

**4) Дополнительные параметры**

– `charset`

– Тот же, что и в параметре `charset` со значением `application/xml`, как в IETF RFC 3923.

**5) Аспекты кодирования**

– Те же, что и для `application/xml`, как в IETF RFC 3923.

**6) Аспекты безопасности**

Объекты типа `samlmetadata+xml` не содержат исполнимого контента. Однако эти объекты являются объектами XML и, следовательно, в них имеются все общие аспекты безопасности, которые представлены в IETF RFC 3923 (раздел 10).

Для того чтобы противостоять возможным проблемам, создатель должен подписать объекты типа `samlmetadata+xml`. Любая такая подпись должна быть заверена получателем данных – и как достоверная подпись, и как подпись отправителя.

**7) Аспекты взаимодействия**

Метаданные SAML явно поддерживают идентификацию протоколов и версий, поддерживаемых идентифицированными элементами. Например, элемент провайдера идентификации может быть обозначен как поддерживающий SAML v2.0 и другие протоколы, если они могут быть однозначно идентифицированы при помощи URI. Информация, поддерживающая этот протокол передается через атрибут `protocolSupportEnumeration` объектов метаданных **RoleDescriptorType**.

**8) Опубликованная Рекомендация**

Метаданные SAML явно определяют использование типа канала передачи MIME `application/samlmetadata+xml`.

Приложения, которые используют этот тип канала передачи:

Потенциально, любое приложение, в котором реализован SAML, а также те приложения, в которых используются спецификации, основанные на SAML.

**9) Дополнительная информация**

**1) Магический(е) номер(а)**

Как правило, они те же самые, что и для `application/xml` в IETF RFC 3023. В частности, корневой элемент XML возвращенного объекта будет иметь имя, определенное в области имен с:

– локальным именем: `EntityDescriptor` или `AffiliationDescriptor` или элементами `Descriptor`;

– областью имен URI: `urn:oasis:names:tc:SAML:2.0:metadata` (область имен метаданных SAMLv2.0).

**10) Расширение(я) файла**

Нет.

**11) Код(ы) файлов Макинтош**

Нет.

**12) Человек и его электронный адрес для контактов для получения информации**

Данная регистрация выполнена от имени Технического комитета OASIS по Службам безопасности (SSTC).

**13) Предназначение**

Общего пользования.

## Дополнение IV

## Использование SSL

Некоторые варианты реализации SAML могут использовать SSL 3.0 в дополнение или в качестве альтернативы TLS 1.0. Варианты реализации, в которых используется SSL 3.0 , должны гарантировать, что общая безопасность данного варианта реализации соответствует ограничениям, наложенным на использование шифров в TLS. Например, требование использовать модуль шифрования TLS_RSA_WITH_3DES_EDE_CBC_SHA переводится как использование модуля шифрования SSL_RSA_WITH_3DES_EDE_CBC_SH. Варианты реализации FIPS, способные работать с SSL, используют модуль шифрования FIPS, соответствующий модулю шифрования SSL SSL_RSA_WITH_3DES_EDE_CBC_SH.

Варианты реализации TLS профиля SAML SSO для Веб, который поддерживает модуль шифрования TLS_RSA_WITH_3DES_EDE_CBC_SH, будет использовать модуль шифрования SSL_RSA_WITH_3DES_EDE_CBC_SH.

## Дополнение V

## Схема SAML – Контекст аутентификации

В настоящем Дополнении содержится Схема SAML Правил аутентификации для сертификата SSL (sslcert).

```
<?xml version="1.0" encoding="UTF-8"?>

<xs:schema targetNamespace="urn:oasis:names:tc:SAML:2.0:ac:classes:TLSClient"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns="urn:oasis:names:tc:SAML:2.0:ac:classes:TLSClient"
  finalDefault="extension"
  blockDefault="substitution"
  version="2.0">

  <xs:redefine schemaLocation="saml-schema-authn-context-types-2.0.xsd">

    <xs:annotation>
      <xs:documentation>
```

```
        Class identifier: urn:oasis:names:tc:SAML:2.0:ac:classes:TLSClient
        Document identifier: saml-schema-authn-context-sslcert-2.0
        Location: http://docs.oasis-open.org/security/saml/v2.0/
        Revision history:
          V2.0 (March, 2005):
            New authentication_u99 context class schema for SAML V2.0 .
      </xs:documentation>
    </xs:annotation>

    <xs:complexType name="AuthnContextDeclarationBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthnContextDeclarationBaseType">
          <xs:sequence>
            <xs:element ref="Identification" minOccurs="0"/>
            <xs:element ref="TechnicalProtection" minOccurs="0"/>
            <xs:element ref="OperationalProtection" minOccurs="0"/>
            <xs:element ref="AuthnMethod"/>
            <xs:element ref="GoverningAgreements" minOccurs="0"/>
            <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
          </xs:sequence>
          <xs:attribute name="ID" type="xs:ID" use="optional"/>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="AuthnMethodBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthnMethodBaseType">
          <xs:sequence>
            <xs:element ref="PrincipalAuthenticationMechanism"/>
            <xs:element ref="Authenticator"/>
            <xs:element ref="AuthenticatorTransportProtocol" minOccurs="0"/>
            <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="PrincipalAuthenticationMechanismType">
      <xs:complexContent>
        <xs:restriction base="PrincipalAuthenticationMechanismType">
          <xs:sequence>
            <xs:element ref="RestrictedPassword"/>
          </xs:sequence>
          <xs:attribute name="preauth" type="xs:integer" use="optional"/>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="AuthenticatorBaseType">
      <xs:complexContent>
        <xs:restriction base="AuthenticatorBaseType">
          <xs:sequence>
            <xs:element ref="DigSig"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="PublicKeyType">
      <xs:complexContent>
        <xs:restriction base="PublicKeyType">
          <xs:attribute name="keyValidation" type="xs:anyURI"
fixed="urn:oasis:names:tc:SAML:2.0:ac:classes:X509"/>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="AuthenticatorTransportProtocolType">
      <xs:complexContent>
        <xs:restriction base="AuthenticatorTransportProtocolType">
```

```
          <xs:sequence>
            <xs:choice>
              <xs:element ref="SSL"/>
              <xs:element ref="WTLS"/>
            </xs:choice>
            <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

  </xs:redefine>

</xs:schema>
```

## Дополнение VI

## Схема XML – Контекст аутентификации для типов

В настоящем дополнении приводится полный листинг схемы XML – Контекст аутентификации для типов и сама схема XML – Контекст аутентификации, используемые для проверки индивидуально обобщенных заявлений. Схема типов сама по себе не имеет целевой области имен и поэтому включена в Дополнение V.

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  elementFormDefault="qualified"
  version="2.0">

  <xs:annotation>
    <xs:documentation>
      Document identifier: saml-schema-authn-context-types-2.0
      Location: http://docs.oasis-open.org/security/saml/v2.0/
      Revision history:
          V2.0 (March, 2005):
          New core authentication context schema types for SAML V2.0 .
    </xs:documentation>
  </xs:annotation>

  <xs:element name="AuthenticationContextDeclaration"
type="AuthnContextDeclarationBaseType">
    <xs:annotation>
      <xs:documentation>
        A particular assertion on an identity
        provider's part with respect to the authentication
        context associated with an authentication assertion.
      </xs:documentation>
    </xs:annotation>
  </xs:element>

  <xs:element name="Identification" type="IdentificationType">
    <xs:annotation>
      <xs:documentation>
        Refers to those characteristics that describe the
        processes and mechanisms
        the Authentication Authority uses to initially create
        an association between a Principal
        and the identity (or name) by which the Principal will
        be known
      </xs:documentation>
    </xs:annotation>
  </xs:element>

  <xs:element name="PhysicalVerification">
    <xs:annotation>
```

```
    <xs:documentation>
      This element indicates that identification has been
      performed in a physical
      face-to-face meeting with the principal and not in an
      online manner.
    </xs:documentation>
  </xs:annotation>
  <xs:complexType>
    <xs:attribute name="credentialLevel">
      <xs:simpleType>
        <xs:restriction base="xs:NMTOKEN">
          <xs:enumeration value="primary"/>
          <xs:enumeration value="secondary"/>
        </xs:restriction>
      </xs:simpleType>
    </xs:attribute>
  </xs:complexType>
</xs:element>

<xs:element name="WrittenConsent" type="ExtensionOnlyType"/>

<xs:element name="TechnicalProtection" type="TechnicalProtectionBaseType">
  <xs:annotation>
    <xs:documentation>
      Refers to those characterstics that describe how the
      'secret' (the knowledge or possession
      of which allows the Principal to authenticate to the
      Authentication Authority) is kept secure
    </xs:documentation>
  </xs:annotation>
</xs:element>

<xs:element name="SecretKeyProtection" type="SecretKeyProtectionType">
  <xs:annotation>
    <xs:documentation>
      This element indicates the types and strengths of
      facilities
      of a UA used to protect a shared secret key from
      unauthorized access and/or use.
    </xs:documentation>
  </xs:annotation>
</xs:element>

<xs:element name="PrivateKeyProtection" type="PrivateKeyProtectionType">
  <xs:annotation>
    <xs:documentation>
      This element indicates the types and strengths of
      facilities
      of a UA used to protect a private key from
      unauthorized access and/or use.
    </xs:documentation>
  </xs:annotation>
</xs:element>

<xs:element name="KeyActivation" type="KeyActivationType">
  <xs:annotation>
    <xs:documentation>The actions that must be performed
      before the private key can be used. </xs:documentation>
  </xs:annotation>
</xs:element>

<xs:element name="KeySharing" type="KeySharingType">
  <xs:annotation>
    <xs:documentation>Whether or not the private key is shared
      with the certificate authority.</xs:documentation>
  </xs:annotation>
</xs:element>

<xs:element name="KeyStorage" type="KeyStorageType">
  <xs:annotation>
    <xs:documentation>
```

```
      In which medium is the key stored.
      memory - the key is stored in memory.
      smartcard - the key is stored in a smartcard.
      token - the key is stored in a hardware token.
      MobileDevice - the key is stored in a mobile device.
      MobileAuthCard - the key is stored in a mobile
      authentication card.
    </xs:documentation>
  </xs:annotation>
</xs:element>

<xs:element name="SubscriberLineNumber" type="ExtensionOnlyType"/>
<xs:element name="UserSuffix" type="ExtensionOnlyType"/>

<xs:element name="Password" type="PasswordType">
  <xs:annotation>
    <xs:documentation>
      This element indicates that a password (or passphrase)
      has been used to
      authenticate the Principal to a remote system.
    </xs:documentation>
  </xs:annotation>
</xs:element>

<xs:element name="ActivationPin" type="ActivationPinType">
  <xs:annotation>
    <xs:documentation>
      This element indicates that a Pin (Personal
      Identification Number) has been used to authenticate the Principal
      to some local system in order to activate a key.
    </xs:documentation>
  </xs:annotation>
</xs:element>

<xs:element name="Token" type="TokenType">
  <xs:annotation>
    <xs:documentation>
      This element indicates that a hardware or software
      token is used
      as a method of identifying the Principal.
    </xs:documentation>
  </xs:annotation>
</xs:element>

<xs:element name="TimeSyncToken" type="TimeSyncTokenType">
  <xs:annotation>
    <xs:documentation>
      This element indicates that a time synchronization
      token is used to identify the Principal. hardware -
      the time synchonization
      token has been implemented in hardware. software - the
      time synchronization
      token has been implemented in software. SeedLength -
      the length, in bits, of the
      random seed used in the time synchronization token.
    </xs:documentation>
  </xs:annotation>
</xs:element>

<xs:element name="Smartcard" type="ExtensionOnlyType">
  <xs:annotation>
    <xs:documentation>
      This element indicates that a smartcard is used to
      identify the Principal.
    </xs:documentation>
  </xs:annotation>
</xs:element>

<xs:element name="Length" type="LengthType">
  <xs:annotation>
    <xs:documentation>
```

```
        This element indicates the minimum and/or maximum
        ASCII length of the password which is enforced (by the UA or the
        IdP). In other words, this is the minimum and/or maximum number of
        ASCII characters required to represent a valid password.
        min - the minimum number of ASCII characters required
        in a valid password, as enforced by the UA or the IdP.
        max - the maximum number of ASCII characters required
        in a valid password, as enforced by the UA or the IdP.
      </xs:documentation>
    </xs:annotation>
  </xs:element>

  <xs:element name="ActivationLimit" type="ActivationLimitType">
    <xs:annotation>
      <xs:documentation>
        This element indicates the length of time for which an
        PIN-based authentication is valid.
      </xs:documentation>
    </xs:annotation>
  </xs:element>

  <xs:element name="Generation">
    <xs:annotation>
      <xs:documentation>
        Indicates whether the password was chosen by the
        Principal or auto-supplied by the Authentication Authority.
        principalchosen - the Principal is allowed to choose
        the value of the password. This is true even if
        the initial password is chosen at random by the UA or
        the IdP and the Principal is then free to change
        the password.
        automatic - the password is chosen by the UA or the
        IdP to be cryptographically strong in some sense,
        or to satisfy certain password rules, and that the
        Principal is not free to change it or to choose a new password.
      </xs:documentation>
    </xs:annotation>

    <xs:complexType>
      <xs:attribute name="mechanism" use="required">
        <xs:simpleType>
          <xs:restriction base="xs:NMTOKEN">
            <xs:enumeration value="principalchosen"/>
            <xs:enumeration value="automatic"/>
          </xs:restriction>
        </xs:simpleType>
      </xs:attribute>
    </xs:complexType>
  </xs:element>

  <xs:element name="AuthnMethod" type="AuthnMethodBaseType">
    <xs:annotation>
      <xs:documentation>
        Refers to those characteristics that define the
        mechanisms by which the Principal authenticates to the
        Authentication Authority.
      </xs:documentation>
    </xs:annotation>
  </xs:element>

  <xs:element name="PrincipalAuthenticationMechanism"
type="PrincipalAuthenticationMechanismType">
    <xs:annotation>
      <xs:documentation>
        The method that a Principal employs to perform
        authentication to local system components.
      </xs:documentation>
    </xs:annotation>
  </xs:element>

  <xs:element name="Authenticator" type="AuthenticatorBaseType">
```

```
    <xs:annotation>
      <xs:documentation>
        The method applied to validate a principal's
        authentication across a network
      </xs:documentation>
    </xs:annotation>
  </xs:element>

  <xs:element name="ComplexAuthenticator" type="ComplexAuthenticatorType">
    <xs:annotation>
      <xs:documentation>
        Supports Authenticators with nested combinations of
        additional complexity.
      </xs:documentation>
    </xs:annotation>
  </xs:element>

  <xs:element name="PreviousSession" type="ExtensionOnlyType">
    <xs:annotation>
      <xs:documentation>
        Indicates that the Principal has been strongly
        authenticated in a previous session during which the IdP has set a
        cookie in the UA. During the present session the Principal has only
        been authenticated by the UA returning the cookie to the IdP.
      </xs:documentation>
    </xs:annotation>
  </xs:element>

  <xs:element name="ResumeSession" type="ExtensionOnlyType">
    <xs:annotation>
      <xs:documentation>
        Rather like PreviousSession but using stronger
        security. A secret that was established in a previous session with
        the Authentication Authority has been cached by the local system
        and is now re-used (e.g. a Master Secret is used to derive new
        session keys in TLS, WTLS).
      </xs:documentation>
    </xs:annotation>
  </xs:element>

  <xs:element name="ZeroKnowledge" type="ExtensionOnlyType">
    <xs:annotation>
      <xs:documentation>
        This element indicates that the Principal has been
        authenticated by a zero knowledge technique as specified in ISO/IEC
        9798-5.
      </xs:documentation>
    </xs:annotation>
  </xs:element>

  <xs:element name="SharedSecretChallengeResponse"
type="SharedSecretChallengeResponseType"/>

  <xs:complexType name="SharedSecretChallengeResponseType">
    <xs:annotation>
      <xs:documentation>
        This element indicates that the Principal has been
        authenticated by a challenge-response protocol utilizing shared
        secret keys and symmetric cryptography.
      </xs:documentation>
    </xs:annotation>
    <xs:sequence>
      <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:attribute name="method" type="xs:anyURI" use="optional"/>
  </xs:complexType>

  <xs:element name="DigSig" type="PublicKeyType">
    <xs:annotation>
      <xs:documentation>
        This element indicates that the Principal has been
```

```
        authenticated by a mechanism which involves the Principal computing
        a digital signature over at least challenge data provided
        by the IdP.
      </xs:documentation>
    </xs:annotation>
  </xs:element>

  <xs:element name="AsymmetricDecryption" type="PublicKeyType">
    <xs:annotation>
      <xs:documentation>
        The local system has a private key but it is used
        in decryption mode, rather than signature mode. For example, the
        Authentication Authority generates a secret and encrypts it using
        the local system's public key: the local system then proves it has
        decrypted the secret.
      </xs:documentation>
    </xs:annotation>
  </xs:element>

  <xs:element name="AsymmetricKeyAgreement" type="PublicKeyType">
    <xs:annotation>
      <xs:documentation>
        The local system has a private key and uses it for
        shared secret key agreement with the Authentication Authority
        (For example, via Diffie Helman).
      </xs:documentation>
    </xs:annotation>
  </xs:element>

  <xs:complexType name="PublicKeyType">
    <xs:sequence>
      <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:attribute name="keyValidation" use="optional"/>
  </xs:complexType>

  <xs:element name="IPAddress" type="ExtensionOnlyType">
    <xs:annotation>
      <xs:documentation>
        This element indicates that the Principal has been
        authenticated through connection from a particular IP address.
      </xs:documentation>
    </xs:annotation>
  </xs:element>

  <xs:element name="SharedSecretDynamicPlaintext" type="ExtensionOnlyType">
    <xs:annotation>
      <xs:documentation>
        The local system and Authentication Authority
        share a secret key. The local system uses this to encrypt a
        randomised string to pass to the Authentication Authority.
      </xs:documentation>
    </xs:annotation>
  </xs:element>

  <xs:element name="AuthenticatorTransportProtocol"
type="AuthenticatorTransportProtocolType">
    <xs:annotation>
      <xs:documentation>
        The protocol across which Authenticator information is
        transferred to an Authentication Authority verifier.
      </xs:documentation>
    </xs:annotation>
  </xs:element>

  <xs:element name="HTTP" type="ExtensionOnlyType">
    <xs:annotation>
      <xs:documentation>
        This element indicates that the Authenticator has been
        transmitted using bare HTTP utilizing no additional security
        protocols.
```

```
    </xs:documentation>
  </xs:annotation>
</xs:element>

<xs:element name="IPSec" type="ExtensionOnlyType">
  <xs:annotation>
    <xs:documentation>
      This element indicates that the Authenticator has been
      transmitted using a transport mechanism protected by an IPSEC session.
    </xs:documentation>
  </xs:annotation>
</xs:element>

<xs:element name="WTLS" type="ExtensionOnlyType">
  <xs:annotation>
    <xs:documentation>
      This element indicates that the Authenticator has been
      transmitted using a transport mechanism protected by a WTLS session.
    </xs:documentation>
  </xs:annotation>
</xs:element>

<xs:element name="MobileNetworkNoEncryption" type="ExtensionOnlyType">
  <xs:annotation>
    <xs:documentation>
      This element indicates that the Authenticator has been
      transmitted solely across a mobile network using no additional
      security mechanism.
    </xs:documentation>
  </xs:annotation>
</xs:element>

<xs:element name="MobileNetworkRadioEncryption" type="ExtensionOnlyType"/>
<xs:element name="MobileNetworkEndToEndEncryption" type="ExtensionOnlyType"/>

<xs:element name="SSL" type="ExtensionOnlyType">
  <xs:annotation>
    <xs:documentation>
      This element indicates that the Authenticator has been
      transmitted using a transport mechanism protected by an SSL or TLS
      session.
    </xs:documentation>
  </xs:annotation>
</xs:element>

<xs:element name="PSTN" type="ExtensionOnlyType"/>
<xs:element name="ISDN" type="ExtensionOnlyType"/>
<xs:element name="ADSL" type="ExtensionOnlyType"/>

<xs:element name="OperationalProtection" type="OperationalProtectionType">
  <xs:annotation>
    <xs:documentation>
      Refers to those characteristics that describe
      procedural security controls employed by the Authentication Authority.
    </xs:documentation>
  </xs:annotation>
</xs:element>

<xs:element name="SecurityAudit" type="SecurityAuditType"/>
<xs:element name="SwitchAudit" type="ExtensionOnlyType"/>
<xs:element name="DeactivationCallCenter" type="ExtensionOnlyType"/>

<xs:element name="GoverningAgreements" type="GoverningAgreementsType">
  <xs:annotation>
    <xs:documentation>
      Provides a mechanism for linking to external (likely
      human readable) documents in which additional business agreements,
      (For example, liability constraints, obligations, etc.) can be placed.
    </xs:documentation>
  </xs:annotation>
</xs:element>
```

```
<xs:element name="GoverningAgreementRef" type="GoverningAgreementRefType"/>

<xs:simpleType name="nymType">
  <xs:restriction base="xs:NMTOKEN">
    <xs:enumeration value="anonymity"/>
    <xs:enumeration value="verinymity"/>
    <xs:enumeration value="pseudonymity"/>
  </xs:restriction>
</xs:simpleType>

<xs:complexType name="AuthnContextDeclarationBaseType">
  <xs:sequence>
    <xs:element ref="Identification" minOccurs="0"/>
    <xs:element ref="TechnicalProtection" minOccurs="0"/>
    <xs:element ref="OperationalProtection" minOccurs="0"/>
    <xs:element ref="AuthnMethod" minOccurs="0"/>
    <xs:element ref="GoverningAgreements" minOccurs="0"/>
    <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
  </xs:sequence>
  <xs:attribute name="ID" type="xs:ID" use="optional"/>
</xs:complexType>

<xs:complexType name="IdentificationType">
  <xs:sequence>
    <xs:element ref="PhysicalVerification" minOccurs="0"/>
    <xs:element ref="WrittenConsent" minOccurs="0"/>
    <xs:element ref="GoverningAgreements" minOccurs="0"/>
    <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
  </xs:sequence>
  <xs:attribute name="nym" type="nymType">
    <xs:annotation>
      <xs:documentation>
        This attribute indicates whether or not the
        Identification mechanisms allow the actions of the Principal to
        be linked to an actual end user.
      </xs:documentation>
    </xs:annotation>
  </xs:attribute>
</xs:complexType>

<xs:complexType name="TechnicalProtectionBaseType">
  <xs:sequence>
    <xs:choice minOccurs="0">
      <xs:element ref="PrivateKeyProtection"/>
      <xs:element ref="SecretKeyProtection"/>
    </xs:choice>
    <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>

<xs:complexType name="OperationalProtectionType">
  <xs:sequence>
    <xs:element ref="SecurityAudit" minOccurs="0"/>
    <xs:element ref="DeactivationCallCenter" minOccurs="0"/>
    <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>

<xs:complexType name="AuthnMethodBaseType">
  <xs:sequence>
    <xs:element ref="PrincipalAuthenticationMechanism" minOccurs="0"/>
    <xs:element ref="Authenticator" minOccurs="0"/>
    <xs:element ref="AuthenticatorTransportProtocol" minOccurs="0"/>
    <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>

<xs:complexType name="GoverningAgreementsType">
  <xs:sequence>
    <xs:element ref="GoverningAgreementRef" maxOccurs="unbounded"/>
```

```
  </xs:sequence>
</xs:complexType>

<xs:complexType name="GoverningAgreementRefType">
  <xs:attribute name="governingAgreementRef" type="xs:anyURI" use="required"/>
</xs:complexType>

<xs:complexType name="PrincipalAuthenticationMechanismType">
  <xs:sequence>
    <xs:element ref="Password" minOccurs="0"/>
    <xs:element ref="RestrictedPassword" minOccurs="0"/>
    <xs:element ref="Token" minOccurs="0"/>
    <xs:element ref="Smartcard" minOccurs="0"/>
    <xs:element ref="ActivationPin" minOccurs="0"/>
    <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
  </xs:sequence>
  <xs:attribute name="preauth" type="xs:integer" use="optional"/>
</xs:complexType>

<xs:group name="AuthenticatorChoiceGroup">
  <xs:choice>
    <xs:element ref="PreviousSession"/>
    <xs:element ref="ResumeSession"/>
    <xs:element ref="DigSig"/>
    <xs:element ref="Password"/>
    <xs:element ref="RestrictedPassword"/>
    <xs:element ref="ZeroKnowledge"/>
    <xs:element ref="SharedSecretChallengeResponse"/>
    <xs:element ref="SharedSecretDynamicPlaintext"/>
    <xs:element ref="IPAddress"/>
    <xs:element ref="AsymmetricDecryption"/>
    <xs:element ref="AsymmetricKeyAgreement"/>
    <xs:element ref="SubscriberLineNumber"/>
    <xs:element ref="UserSuffix"/>
    <xs:element ref="ComplexAuthenticator"/>
  </xs:choice>
</xs:group>

<xs:group name="AuthenticatorSequenceGroup">
  <xs:sequence>
    <xs:element ref="PreviousSession" minOccurs="0"/>
    <xs:element ref="ResumeSession" minOccurs="0"/>
    <xs:element ref="DigSig" minOccurs="0"/>
    <xs:element ref="Password" minOccurs="0"/>
    <xs:element ref="RestrictedPassword" minOccurs="0"/>
    <xs:element ref="ZeroKnowledge" minOccurs="0"/>
    <xs:element ref="SharedSecretChallengeResponse" minOccurs="0"/>
    <xs:element ref="SharedSecretDynamicPlaintext" minOccurs="0"/>
    <xs:element ref="IPAddress" minOccurs="0"/>
    <xs:element ref="AsymmetricDecryption" minOccurs="0"/>
    <xs:element ref="AsymmetricKeyAgreement" minOccurs="0"/>
    <xs:element ref="SubscriberLineNumber" minOccurs="0"/>
    <xs:element ref="UserSuffix" minOccurs="0"/>
    <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
  </xs:sequence>
</xs:group>

<xs:complexType name="AuthenticatorBaseType">
  <xs:sequence>
    <xs:group ref="AuthenticatorChoiceGroup"/>
    <xs:group ref="AuthenticatorSequenceGroup"/>
  </xs:sequence>
</xs:complexType>

<xs:complexType name="ComplexAuthenticatorType">
  <xs:sequence>
    <xs:group ref="AuthenticatorChoiceGroup"/>
    <xs:group ref="AuthenticatorSequenceGroup"/>
  </xs:sequence>
</xs:complexType>
```

```
  <xs:complexType name="AuthenticatorTransportProtocolType">
    <xs:sequence>
      <xs:choice minOccurs="0">
        <xs:element ref="HTTP"/>
        <xs:element ref="SSL"/>
        <xs:element ref="MobileNetworkNoEncryption"/>
        <xs:element ref="MobileNetworkRadioEncryption"/>
        <xs:element ref="MobileNetworkEndToEndEncryption"/>
        <xs:element ref="WTLS"/>
        <xs:element ref="IPSec"/>
        <xs:element ref="PSTN"/>
        <xs:element ref="ISDN"/>
        <xs:element ref="ADSL"/>
      </xs:choice>
      <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>

  <xs:complexType name="KeyActivationType">
    <xs:sequence>
      <xs:element ref="ActivationPin" minOccurs="0"/>
      <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>

  <xs:complexType name="KeySharingType">
    <xs:attribute name="sharing" type="xs:boolean" use="required"/>
  </xs:complexType>

  <xs:complexType name="PrivateKeyProtectionType">
    <xs:sequence>
      <xs:element ref="KeyActivation" minOccurs="0"/>
      <xs:element ref="KeyStorage" minOccurs="0"/>
      <xs:element ref="KeySharing" minOccurs="0"/>
      <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>

  <xs:complexType name="PasswordType">
    <xs:sequence>
      <xs:element ref="Length" minOccurs="0"/>
      <xs:element ref="Alphabet" minOccurs="0"/>
      <xs:element ref="Generation" minOccurs="0"/>
      <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:attribute name="ExternalVerification" type="xs:anyURI" use="optional"/>
  </xs:complexType>

  <xs:element name="RestrictedPassword" type="RestrictedPasswordType"/>

  <xs:complexType name="RestrictedPasswordType">
    <xs:complexContent>
      <xs:restriction base="PasswordType">
        <xs:sequence>
          <xs:element name="Length" type="RestrictedLengthType" minOccurs="1"/>
          <xs:element ref="Generation" minOccurs="0"/>
          <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
        </xs:sequence>
        <xs:attribute name="ExternalVerification" type="xs:anyURI"
use="optional"/>
      </xs:restriction>
    </xs:complexContent>
  </xs:complexType>

  <xs:complexType name="RestrictedLengthType">
    <xs:complexContent>
      <xs:restriction base="LengthType">
        <xs:attribute name="min" use="required">
          <xs:simpleType>
            <xs:restriction base="xs:integer">
              <xs:minInclusive value="3"/>
```

```
          </xs:restriction>
        </xs:simpleType>
      </xs:attribute>
      <xs:attribute name="max" type="xs:integer" use="optional"/>
    </xs:restriction>
  </xs:complexContent>
</xs:complexType>

<xs:complexType name="ActivationPinType">
  <xs:sequence>
    <xs:element ref="Length" minOccurs="0"/>
    <xs:element ref="Alphabet" minOccurs="0"/>
    <xs:element ref="Generation" minOccurs="0"/>
    <xs:element ref="ActivationLimit" minOccurs="0"/>
    <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>

<xs:element name="Alphabet" type="AlphabetType"/>
<xs:complexType name="AlphabetType">
  <xs:attribute name="requiredChars" type="xs:string" use="required"/>
  <xs:attribute name="excludedChars" type="xs:string" use="optional"/>
  <xs:attribute name="case" type="xs:string" use="optional"/>
</xs:complexType>

<xs:complexType name="TokenType">
  <xs:sequence>
    <xs:element ref="TimeSyncToken"/>
    <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>

<xs:simpleType name="DeviceTypeType">
  <xs:restriction base="xs:NMTOKEN">
    <xs:enumeration value="hardware"/>
    <xs:enumeration value="software"/>
  </xs:restriction>
</xs:simpleType>

<xs:simpleType name="booleanType">
  <xs:restriction base="xs:NMTOKEN">
    <xs:enumeration value="true"/>
    <xs:enumeration value="false"/>
  </xs:restriction>
</xs:simpleType>

<xs:complexType name="TimeSyncTokenType">
  <xs:attribute name="DeviceType" type="DeviceTypeType" use="required"/>
  <xs:attribute name="SeedLength" type="xs:integer" use="required"/>
  <xs:attribute name="DeviceInHand" type="booleanType" use="required"/>
</xs:complexType>

<xs:complexType name="ActivationLimitType">
  <xs:choice>
    <xs:element ref="ActivationLimitDuration"/>
    <xs:element ref="ActivationLimitUsages"/>
    <xs:element ref="ActivationLimitSession"/>
  </xs:choice>
</xs:complexType>

<xs:element name="ActivationLimitDuration" type="ActivationLimitDurationType">
  <xs:annotation>
    <xs:documentation>
      This element indicates that the Key Activation Limit is
      defined as a specific duration of time.
    </xs:documentation>
  </xs:annotation>
</xs:element>

<xs:element name="ActivationLimitUsages" type="ActivationLimitUsagesType">
  <xs:annotation>
```

```
    <xs:documentation>
      This element indicates that the Key Activation Limit is
      defined as a number of usages.
    </xs:documentation>
  </xs:annotation>
</xs:element>

<xs:element name="ActivationLimitSession" type="ActivationLimitSessionType">
  <xs:annotation>
    <xs:documentation>
      This element indicates that the Key Activation Limit is
      the session.
    </xs:documentation>
  </xs:annotation>
</xs:element>

<xs:complexType name="ActivationLimitDurationType">
  <xs:attribute name="duration" type="xs:duration" use="required"/>
</xs:complexType>

<xs:complexType name="ActivationLimitUsagesType">
  <xs:attribute name="number" type="xs:integer" use="required"/>
</xs:complexType>

<xs:complexType name="ActivationLimitSessionType"/>

<xs:complexType name="LengthType">
  <xs:attribute name="min" type="xs:integer" use="required"/>
  <xs:attribute name="max" type="xs:integer" use="optional"/>
</xs:complexType>

<xs:simpleType name="mediumType">
  <xs:restriction base="xs:NMTOKEN">
    <xs:enumeration value="memory"/>
    <xs:enumeration value="smartcard"/>
    <xs:enumeration value="token"/>
    <xs:enumeration value="MobileDevice"/>
    <xs:enumeration value="MobileAuthCard"/>
  </xs:restriction>
</xs:simpleType>

<xs:complexType name="KeyStorageType">
  <xs:attribute name="medium" type="mediumType" use="required"/>
</xs:complexType>

<xs:complexType name="SecretKeyProtectionType">
  <xs:sequence>
    <xs:element ref="KeyActivation" minOccurs="0"/>
    <xs:element ref="KeyStorage" minOccurs="0"/>
    <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>

<xs:complexType name="SecurityAuditType">
  <xs:sequence>
    <xs:element ref="SwitchAudit" minOccurs="0"/>
    <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>

<xs:complexType name="ExtensionOnlyType">
  <xs:sequence>
    <xs:element ref="Extension" minOccurs="0"  maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>

<xs:element name="Extension" type="ExtensionType"/>

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

```
    </xs:sequence>
  </xs:complexType>

</xs:schema>


<?xml version="1.0" encoding="UTF-8"?>
<xs:schema
  targetNamespace="urn:oasis:names:tc:SAML:2.0:ac"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns="urn:oasis:names:tc:SAML:2.0:ac"
  blockDefault="substitution"
  version="2.0">

  <xs:annotation>
    <xs:documentation>
      Document identifier: saml-schema-authn-context-2.0
      Location: http://docs.oasis-open.org/security/saml/v2.0/
      Revision history:
        V2.0 (March, 2005):
          New core authentication context schema for SAML V2.0 .
          This is just an include of all types from the schema
          referred to in the include statement below.
    </xs:documentation>
  </xs:annotation>

  <xs:include schemaLocation="saml-schema-authn-context-types-2.0.xsd"/>

</xs:schema>
```

ПРИМЕЧАНИЕ. – Использование SSL представлено в Дополнении IV.

## Дополнение VII

## Профиль атрибута SAML DCE PAC

В настоящем дополнении рассматривается профиль связи SAML для распределенной среды вычислений (DCE), сертификаты атрибута привилегий (PAC) (см. открытый источник DCE).

### VII.1 Профиль атрибута DCE PAC

Профиль атрибута DCE PAC определяет выражение информации DCE PAC, в виде имен и значений атрибута SAML. Он используется для стандартизации преобразования между первичной информацией, которая формирует идентификатор DCE клиента, и множеством атрибутов SAML. Этот профиль построен на базе профиля атрибута UUID, определенного в 11.4.9.3.

1) **Требуемая информация**

   - **Идентификация**: `urn:oasis:names:tc:SAML:2.0:profiles:attribute:DCE` (это также целевая область имен, назначенная в соответствующей схеме профиля атрибута DCE PAC в Приложении A)
   - **Контактная информация**: security-services-comment@lists.oasis-open.org
   - **Описание**: Приведено ниже.
   - **Обновления**: Нет.

2) **Описание PAC**

   DCE PAC – это расширяемая структура, которая передает произвольные атрибуты регистра DCE, но корневое множество данных является общим для всех клиентов и составляет объем идентификаторов DCE:

   - DCE клиента "элемент" или "ячейка";
   - уникальный идентификатор клиента;
   - участие клиента в первичной локальной группе DCE;
   - варианты участия клиента в локальной группе DCE (многозначный);

- варианты участия клиента в иностранной группе DCE (многозначный).

Основным значением каждого из этих атрибутов является UUID.

**3) Система обозначений атрибутов SAML**

Настоящий профиль определяет преобразование конкретной информации DCE в атрибуты SAML и, следовательно, определяет действительные конкретные названия атрибутов, а не условные обозначения имен.

Для всех атрибутов, определенных этим профилем, атрибут XML `NameFormat` в элементах `<Attribute>` должен имеет значение `urn:oasis:names:tc:SAML:2.0:attrname-format:uri`.

Для обеспечения удобочитаемости для человека, для некоторых приложений может быть установлено требование передавать вместе с URI дополнительную строку. Для этой цели может использоваться дополнительный атрибут XML `FriendlyName`.

**4) Сравнение имени атрибута**

Два элемента `<Attribute>` обозначают один и тот же атрибут SAML, если и только если значения их атрибутов XML `Name` равны в том смысле, как это описано в Рекомендации МСЭ-Т X.667. Атрибут `FriendlyName` не играет при сравнении никакой роли.

**5) Определяемые профилем атрибуты XML**

Никаких дополнительных атрибутов XML не определяется для использования с элементом `<Attribute>`.

**6) Значения атрибутов SAML**

Первичным(и) значением(ями) для каждого атрибута, определенного этим профилем, является UUID. Для представления таких значений используемся синтаксис URN, описанный в 11.4.9.3.

Однако этот профиль допускает наличие дополнительной информации, связанной со значением UUID, состоящей из дружественной удобочитаемой для человека строки, и дополнительного UUID, представляющего собой элемент или ячейку DCE. Дополнительная информация передается в элементе `<AttributeValue>` атрибутов XML `FriendlyName` и `Realm`, определенных в области имен XML `urn:oasis:names:tc:SAML:2.0:profiles:attribute:DCE`. Это не то же самое, что атрибут XML `FriendlyName`, определенный в разделе 8, хотя он имеют общую базовую цель.

Приведенный далее листинг схемы показывает как определяемые профилем атрибуты XML и сложный тип используются в `xsi:type` (Приложение A):

```
<schema targetNamespace="urn:oasis:names:tc:SAML:2.0:profiles:attribute:DCE"
    xmlns:dce="urn:oasis:names:tc:SAML:2.0:profiles:attribute:DCE"
    xmlns="http://www.w3.org/2001/XMLSchema"
    elementFormDefault="unqualified"
    attributeFormDefault="unqualified"
    blockDefault="substitution"
    version="2.0">
    <annotation>
        <documentation>
            Document identifier: saml-schema-dce-2.0
            Location: http://docs.oasis-open.org/security/saml/v2.0/
            Revision history:
            V2.0 (March, 2005):
                Custom schema for DCE attribute profile, first published in
SAML 2.0.
        </documentation>
    </annotation>
    <complexType name="DCEValueType">
        <simpleContent>
            <extension base="anyURI">
                <attribute ref="dce:Realm" use="optional"/>
                <attribute ref="dce:FriendlyName" use="optional"/>
            </extension>
        </simpleContent>
    </complexType>
    <attribute name="Realm" type="anyURI"/>
    <attribute name="FriendlyName" type="string"/>
</schema>
```

**7) Атрибут Определения**

Далее показано множество атрибутов SAML, определенных этим профилем. В каждом случае атрибут XML `xsi:type` может быть включен в элемент `<AttributeValue>`, но должен иметь значение **dce:DCEValueType**, где префикс dce является произвольным и должен быть привязан к области имен XML `urn:oasis:names:tc:SAML:2.0:profiles:attribute:DCE`.

Такое использование `xsi:type` требует, чтобы потребители, проверяющие атрибут, включали схему расширения, определенную этим профилем.

**a) Элемент**

Этот атрибут с одним значением представляет собой подтверждение SAML для элемента или ячейки DCE объекта.

**Имя**: `urn:oasis:names:tc:SAML:2.0:profiles:attribute:DCE:realm`

Один-единственный элемент `<AttributeValue>` содержит UUID в форме URN, идентифицирующий подтверждение SAML для элемента или ячейки DCE объекта, с дополнительным определяемым профилем атрибутом XML `FriendlyName`, содержащим строку названия элемента.

**b) Клиент**

Этот атрибут с одним значением представляет собой подтверждение SAML для идентификатора DCE объекта.

**Имя**: `urn:oasis:names:tc:SAML:2.0:profiles:attribute:DCE:principal`

Один-единственный элемент `<AttributeValue>` содержит UUID в форме URN, идентифицирующий подтверждение SAML для идентификатора DCE клиента, с дополнительным определяемым профилем атрибутом XML `FriendlyName`, содержащим строку названия клиента.

Определяемый профилем атрибут XML `Realm` может быть включен, и должен содержать UUID в форме URN, идентифицирующий подтверждение SAML для элемента/ячейки DCE объекта.

**c) Первичная группа**

Этот атрибут с одним значением представляет собой подтверждение SAML для участия объекта в первичной группе DCE.

**Имя**: `urn:oasis:names:tc:SAML:2.0:profiles:attribute:DCE:primary-group`

Один-единственный элемент `<AttributeValue>` содержит UUID в форме URN, идентифицирующий подтверждение SAML для первичной группы DCE, с дополнительным определяемым профилем атрибутом XML `FriendlyName`, содержащим строку названия первичной группы.

Определяемый профилем атрибут XML Realm может быть включен, и должен содержать UUID в форме URN, идентифицирующий подтверждение SAML для элемента/ячейки DCE объекта.

**d) Группы**

Этот атрибут с множеством значений представляет собой подтверждение SAML для участия объекта в локальной группе DCE.

**Имя**: `urn:oasis:names:tc:SAML:2.0:profiles:attribute:DCE:groups`

Каждый элемент `<AttributeValue>` содержит UUID в форме URN, идентифицирующий участие в группе DCE объекта подтверждения SAML, с дополнительным определяемым профилем атрибутом XML `FriendlyName`, содержащим строку названия группы.

Определяемый профилем атрибут XML Realm может быть включен, и должен содержать UUID в форме URN, идентифицирующий подтверждение SAML для элемента/ячейки DCE объекта.

**e) Иностранные группы**

Этот атрибут со множеством значений представляет собой подтверждение SAML для участия объекта в иностранной группе DCE

**Имя**: `urn:oasis:names:tc:SAML:2.0:profiles:attribute:DCE:foreign-groups`

Каждый элемент `<AttributeValue>` содержит UUID в форме URN, идентифицирующий участие в иностранной группе DCE объекта подтверждения SAML, с дополнительным определяемым профилем атрибутом XML `FriendlyName`, содержащим строку названия группы.

Определяемый профилем атрибут XML Realm должен быть включен, и должен содержать UUID в форме URN, идентифицирующий подтверждение SAML для элемента/ячейки DCE иностранной группы.

## VII.2 Схема SAML dce

Далее показана Схема SAML – Правила аутентификации для среды распределенных вычислений (DCE).

```
<?xml version="1.0" encoding="UTF-8"?>
<schema targetNamespace="urn:oasis:names:tc:SAML:2.0:profiles:attribute:DCE"
    xmlns:dce="urn:oasis:names:tc:SAML:2.0:profiles:attribute:DCE"
    xmlns="http://www.w3.org/2001/XMLSchema"
```

```
    elementFormDefault="unqualified"
    attributeFormDefault="unqualified"
    blockDefault="substitution"
    version="2.0">
    <annotation>
        <documentation>
            Document identifier: saml-schema-dce-2.0
            Location: http://docs.oasis-open.org/security/saml/v2.0/
            Revision history:
            V2.0 (March, 2005):
                Custom schema for DCE attribute profile, first published in SAML 2.0.
        </documentation>
    </annotation>
    <complexType name="DCEValueType">
        <simpleContent>
            <extension base="anyURI">
                <attribute ref="dce:Realm" use="optional"/>
                <attribute ref="dce:FriendlyName" use="optional"/>
            </extension>
        </simpleContent>
    </complexType>
    <attribute name="Realm" type="anyURI"/>
    <attribute name="FriendlyName" type="string"/>
</schema>
```

## VII.3 Пример

Далее приведен пример преобразования данных PAC в атрибуты SAML, принадлежащие DCE клиента пот имени "jdoe" в элементе "example.com", члену локальных групп "cubicle-dwellers" и "underpaid" и иностранной группы "engineers".

```
<saml:Assertion
xmlns:dce="urn:oasis:names:tc:SAML:2.0:profiles:attribute:DCE" ...>
  <saml:Issuer>...</saml:Issuer>
  <saml:Subject>...</saml:Subject>
  <saml:AttributeStatement>
  <saml:Attribute NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-
format:uri"
 Name="urn:oasis:names:tc:SAML:2.0:profiles:attribute:DCE:realm">
    <saml:AttributeValue xsi:type="dce:DCEValueType"
dce:FriendlyName="example.com">
    urn:uuid:003c6cc1-9ff8-10f9-990f-004005b13a2b
    </saml:AttributeValue>
  </saml:Attribute>
  <saml:Attribute NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-
format:uri"
 Name="urn:oasis:names:tc:SAML:2.0:profiles:attribute:DCE:principal">
    <saml:AttributeValue xsi:type="dce:DCEValueType"
dce:FriendlyName="jdoe">
    urn:uuid:00305ed1-a1bd-10f9-a2d0-004005b13a2b
    </saml:AttributeValue>
  </saml:Attribute>
  <saml:Attribute NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-
format:uri"
 Name="urn:oasis:names:tc:SAML:2.0:profiles:attribute:DCE:primary-group">
    <saml:AttributeValue xsi:type="dce:DCEValueType"
 dce:FriendlyName="cubicle-dwellers">
    urn:uuid:008c6181-a288-10f9-b6d6-004005b13a2b
    </saml:AttributeValue>
  </saml:Attribute>
  <saml:Attribute NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-
format:uri"
 Name="urn:oasis:names:tc:SAML:2.0:profiles:attribute:DCE:groups">
    <saml:AttributeValue xsi:type="dce:DCEValueType"
 dce:FriendlyName="cubicle-dwellers">
    urn:uuid:008c6181-a288-10f9-b6d6-004005b13a2b
    </saml:AttributeValue>
```

```
    <saml:AttributeValue xsi:type="dce:DCEValueType"
dce:FriendlyName="underpaid">
    urn:uuid:006a5a91-a2b7-10f9-824d-004005b13a2b
    </saml:AttributeValue>
  </saml:Attribute>
  <saml:Attribute NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-
format:uri"
 Name="urn:oasis:names:tc:SAML:2.0:profiles:attribute:DCE:foreign-groups">
    <saml:AttributeValue xsi:type="dce:DCEValueType"
dce:FriendlyName="engineers"
 dce:Realm="urn:uuid:00583221-a35f-10f9-8b6e-004005b13a2b">
    urn:uuid:00099cf1-a355-10f9-9e95-004005b13a2b
    </saml:AttributeValue>
  </saml:Attribute>
  </saml:AttributeStatement>
</saml:Assertion>
```

## Дополнение VIII

## Разъяснения OASIS для языка SAML

Настоящее дополнение добавляет обзоры, которые были выполнены для SAML v2.0 в рамках группы OASIS. Группа OASIS SAML решила опубликовать эти разъяснительные комментарии в виде отдельного документа (см. PE OASIS:2006).Эти разъяснения не являются нормативными и не введены в версию 2.0 языка SAML группы OASIS. В настоящей Рекомендации, эти обзоры перечислены в настоящем Дополнение для гарантии итого, что те, кто будет реализовывать SAML, осведомлены относительно обсуждений, которые имели место после опубликования OASIS SAML v2.0 в качестве стандарта OASIS.

### VIII.1 Возможная ошибка: PE14

**Описание: Allowcreate требует более четкого определения**

**Применимость в настоящей Рекомендации**

Обратите внимание на соответствующие примечания в 8.2.4.1 и 8.2.6. Кроме того, ниже приведено разъяснение второго абзаца подраздела 8.2.6.3.

Если в запрос включен элемент `<Terminate>`, то представляющий его провайдер указывает, что (в случае для провайдера услуг) он больше не будет принимать подтверждения от этого провайдера идентификации или (в случае для провайдера идентификации) он больше не будет создавать подтверждения об этом клиенте для этого провайдера услуг.

Если принимающий провайдер поддерживает состояние, связанное с идентификатором имени, например, значение самого идентификатора (в случае парного идентификатора), значение `SPProvidedID`, согласие отправителя создать/использовать идентификатор и т. д., тогда приемник может выполнить любую поддержку с осознанием того, что взаимосвязь, представленная идентификатором имени, завершена.

Любые следующие действия в отношении клиента выполняемые приемником от имени отправителя, (например, последующий запрос `<AuthnRequest>`) должны выполняться способом, соответствующим отсутствию любого предыдущего состояния.

Завершение, потенциально, является этапом очистки для любого состояния функций управления, инициированным применением атрибута `AllowCreate` в протоколе запроса аутентификации, описанном в 8.2.4. Варианты реализации, в которых используется этот атрибут, вероятно, будут избегать использования элемента `<Terminate>` или будут рассматривать его только как рекомендательный.

Отметим, что в большинстве случаев (известным исключением валяются правила, касающиеся атрибута `SPProvidedID`), ни у провайдеров идентификации, ни у провайдеров услуг не существует требований относительно создания или использования постоянного состояния. Следовательно, никакие функции явно не объявлены обязательными, кода принимается элемент `<Terminate>` = 450. Однако если имеется постоянное состояние, относящееся к использованию идентификатора (так, если прикреплен атрибут `SPProvidedID`), элемент `<Terminate>` является явным указанием того, что это состояние должно быть удалено (или отмечено как устаревшее).

## VIII.2 Возможная ошибка: PE26

**Описание: Требуется разъяснение профиля SSO**

**Применимость в настоящей Рекомендации: Далее разъясняются следующие подразделы.**

#### 11.4.1.4.2 Использование <Response>

Если провайдер идентификации желает вернуть ошибку, он не должен включать в сообщение `<Response>` каких-либо подтверждений. В противном случае если запрос выполнен успешно (или если ответ никак не связан с запросом), элемент `<Response>` должен соответствовать нижеследующему описанию.

- Если ответ не подписан, то элемент `<Issuer>` может быть пропущен, но если он представлен (или если ответ подписан) он должен содержать уникальный идентификатор создавшего его провайдера идентификации; атрибут `Format` должен быть пропущен или должен иметь значение `urn:oasis:names:tc:SAML:2.0:nameid-format:entity`.
- Он должен содержать как минимум одно подтверждение `<Assertion>`. Элемент `<Issuer>` каждого подтверждения должен содержать уникальный идентификатор отвечающего провайдера идентификации; атрибут `Format` должен быть пропущен или должен иметь значение `urn:oasis:names:tc:SAML:2.0:nameid-format:entity`. Отметим, что этот профиль предполагает наличие одного отвечающего провайдера идентификации, и все подтверждения в ответе должны быть созданы одни и тем же элементом.
- Если включается несколько подтверждений, то элемент `<Subject>` каждого подтверждения должен указывать на одного и того же клиента. Допускается, чтобы содержание элементов `<Subject>` было различным (например, используя различные `<NameID>` или альтернативные элементы `<SubjectConfirmation>`).
- Любые подтверждения, созданные для использования с применением этого профиля, должны содержать элемент `<Subject>`, в котором имеется как минимум один элемент `<SubjectConfirmation>`, содержащий атрибут `Method = urn:oasis:names:tc:SAML:2.0:cm:bearer`. Такое подтверждение называется подтверждением канала передачи. Подтверждения канала передачи могут содержать дополнительные элементы `<SubjectConfirmation>`.
- Могут также быть включены подтверждения без подтверждения канала передачи `<SubjectConfirmation>`; обработка дополнительных подтверждений или элементов `<SubjectConfirmation>` в данном профиле не определяется.
- Как минимум один элемент `<SubjectConfirmation>` канала передачи должен содержать элемент `<SubjectConfirmationData>`, который сам может содержать атрибут `Recipient`, в состав которого входит URL подтверждения услуги для пользователя, полученного от провайдера услуг, и атрибут `NotOnOrAfter`, который ограничивает временное окно, в течение которого может быть доставлено подтверждение. Он может также содержать атрибут `Address`, ограничивающий адрес клиента, с которого может быть доставлено подтверждение. Он не должен содержать атрибут `NotBefore`. Если содержащее его сообщения является ответом на запрос `<AuthnRequest>`, тогда атрибут `InResponseTo` должен соответствовать `ID` запроса.
- Множество из одного или нескольких подтверждений должно содержать, как минимуму, одно утверждение `<AuthnStatement>`, которое касается аутентификации провайдера идентификации. В него может быть включено несколько элементов `<AuthnStatement>`, но семантика нескольких элементов не включена в данных профиль.
- Если провайдер идентификации поддерживает профиль единого выхода из системы, определенный в 11.4.1.4.5, то любые подтверждения аутентификации должны содержать атрибут `SessionIndex`, позволяющий провайдеру услуг сделать запросы единого выхода из системы еще до сеанса связи.
- В подтверждение(я) канала передачи могут быть включены другие утверждения на усмотрение провайдера идентификации. В частности, могут быть включены элементы `<AttributeStatement>`. Запрос `<AuthnRequest>` может содержать атрибут XML `AttributeConsumingServiceIndex`, указывающий в своем составе желаемые или требуемые атрибуты, описанные в разделе 9. Провайдер идентификации может игнорировать это либо передавать другие атрибуты по своему усмотрению.
- Каждое подтверждение канала передачи должно содержать элемент `<AudienceRestriction>`, включающий уникальный идентификатор провайдера услуг в качестве элемента `<Audience>`.
- Могут быть включены другие условия (и другие элементы `<Audience>`), запрошенные провайдером услуг или по усмотрению провайдера идентификации. (Несомненно, для того чтобы подтверждение считалось достоверным, провайдер услуг должен понимать все эти условия.)
- Провайдер идентификации не обязан в точности выполнять запрошенное множество условий `<Conditions>` в запросе `<AuthnRequest>`, если таковые указаны.

#### 11.4.1.4.3 Правила обработки сообщения <Response>

Вне зависимости от того, какая связь SAML используется, провайдер услуг должен выполнять следующее.

- Проверить все подписи, имеющиеся в подтверждении(ях) или ответе.
- Удостовериться, что атрибут `Recipient` в элементе `<SubjectConfirmationData>` канала передачи соответствует URL подтверждения услуги потребителя, на который был доставлен `<Response>` или артефакт.
- Удостовериться, что срок, указанный в атрибуте `NotOnOrAfter` в элементе канала передачи `<SubjectConfirmationData>`, еще не прошел, с учетом допустимого фазового сдвига синхронизации между провайдерами.
- Удостовериться, что атрибут `InResponseTo` в элементе канала передачи `<SubjectConfirmationData>` равен `ID` в исходном сообщении `<AuthnRequest>`, если только этот ответ не является незапрошенным, в этом случае этот атрибут не должен быть представлен.
- Удостовериться, что все используемые подтверждения являются достоверными в других отношениях. Отметим, что, хотя может быть представлено несколько элементов `<SubjectConfirmation>` канала передачи, успешная оценка каждого такого элемента в отдельности в соответствии с настоящим профилем является достаточной для того, чтобы удостоверить подтверждение. Однако каждое подтверждение, если их представлено несколько, должно оцениваться независимо от других.
- Если элемент `<SubjectConfirmationData>` канала передачи содержит атрибут `Address`, то провайдер услуг может сравнить с ним адрес агента пользователя клиента.
- Любое подтверждение, которое является недостоверным, или для которого не могут быть выполнены требования по удостоверению подлинности объекта подтверждения, должно быть отброшено и не должно использоваться для установления контекста безопасности для клиента.
- Если утверждение `<AuthnStatement>`, используемое для установления контекста безопасности для клиента, содержит атрибут `SessionNotOnOrAfter`, то контекст безопасности должен быть отброшен сразу по достижении времени, указанного в этом атрибуте, если только провайдер услуг не устанавливает повторно идентификацию клиента, повторив использование этого профиля. Отметим, что, если представлено несколько элементов `<AuthnStatement>`, используется значение `SessionNotOnOrAfter`, наиболее близкое к текущему времени.

#### 11.4.1.4.4 Правила обработки, специфичные для связи POST

Если для доставки сообщения `<Response>` используется связь HTTP POST, то каждое подтверждение должно быт защищено цифровой подписью. Это может быть выполнено путем подписания каждого отдельного элемента `<Assertion>` или путем подписания элемента `<Response>`.

Провайдер услуг должен гарантировать, что подтверждения для канала передачи не повторяются, путем сохранения множества использованных значений `ID` на протяжении времени, в течение которого это подтверждение считается достоверным, основываясь на атрибуте `NotOnOrAfter` элемента `<SubjectConfirmationData>`.

# БИБЛИОГРАФИЯ


- **FIPS-197** (2001), *Advanced Encryption Standard* (AES).
- **IETF RFC 1738** (1994), *Uniform Resource Locators (URL)*.
- **IETF RFC 2256** (1997), *A Summary of the X.500 (96) User Schema for use with LDAPv3*.
- **IETF RFC 2279** (1998), *UTF-8, a transformation format of ISO 10646*.
- **IETF RFC 2743** (2000), *Generic Security Service Application Program Interface Version 2, Update 1*.
- **DCE**, *Distributed Computing Environment (DCE)*, Open Source. See http://www.opengroup.org/dce.
- **OASIS Authentication Context 2.0**, *Authentication Context for the OASIS Security Assertion Markup Language (SAML) V2.0*, 15 March 2005, http://www.oasis-open.org/apps/org/workgroup/security/.
- **OASIS Bindings 1**, *Bindings and Profiles for the OASIS Security Assertion Markup Language (SAML),* 5 November 2002, http://www.oasis-open.org/apps/org/workgroup/security/.
- **OASIS Bindings 1.1**, *Bindings and Profiles for the OASIS Security Assertion Markup Language (SAML),* 22 September 2003, http://www.oasis-open.org/apps/org/workgroup/security/.
- **OASIS Bindings 2.0**, *Bindings for the OASIS Security Assertion Markup Language (SAML) V2.0*, 15 March 2005, http://www.oasis-open.org/apps/org/workgroup/security/.
- **OASIS Conformance 2.0**, *Conformance Requirements for the OASIS Security Assertion Markup Language (SAML) V2.00*, 15 March 2005, http://www.oasis-open.org/apps/org/workgroup/security/.
- **OASIS Glossary 2.0**, *Glossary for the OASIS Security Assertion Markup Language (SAML) V2.0*, 15 March 2005, http://www.oasis-open.org/apps/org/workgroup/security/.
- **OASIS Metadata 2.0**, *Metadata for the OASIS Security Assertion Markup Language (SAML) V2.0*, 15 March 2005, http://www.oasis-open.org/apps/org/workgroup/security/.
- **OASIS Errata Document 24**, *Revision 24 draft of the non-normative SAML V2.0 Errata document*, 27 February 2006, http://www.oasis-open.org/committees/download.php/16935/sstc-saml-errata-2.0-draft-24.pdf.
- **OASIS Protocol 1.0**, *Assertions and Protocol for the OASIS Security Assertion Markup Language (SAML),* 5 November 2002, http://www.oasis-open.org/apps/org/workgroup/security/.
- **OASIS Protocol 1.1**, *Assertions and Protocol for the OASIS Security Assertion Markup Language (SAML),* 22 September 2003, http://www.oasis-open.org/apps/org/workgroup/security/.
- **OASIS Protocol 2.0**, *Profiles for the OASIS Security Assertion Markup Language (SAML) V2.0*, 15 March 2005, http://www.oasis-open.org/apps/org/workgroup/security/.
- **OASIS SAML 1.0**, *Security Assertion Markup Language (SAML) Version 1.0 Specification Set,* 5 November 2002, http://www.oasis-open.org/apps/org/workgroup/security/.
- **OASIS SAML 1.1**, *Security Assertion Markup Language (SAML) Version 1.1 Specification Set,* 22 September 2003, http://www.oasis-open.org/apps/org/workgroup/security/.
- **OASIS Security 1**, *Security Considerations for the OASIS Security Assertion Markup Language (SAML),* 5 November 2002, http://www.oasis-open.org/apps/org/workgroup/security/.
- **OASIS Security 1.1**, *Security Considerations for the OASIS Security Assertion Markup Language (SAML),* 22 September 2003, http://www.oasis-open.org/apps/org/workgroup/security/.
- **OASIS Security 2.0**, *Security and Privacy Considerations for the OASIS Security Assertion Markup Language (SAML) V2.0*, 15 March 2005, http://www.oasis-open.org/apps/org/workgroup/security/.
- **OASIS XACML1-1**, *eXtensible Access Control Markup Language (XACML) V1.1,* 24 July 2003, http://www.oasis-open.org/apps/org/workgroup/xacml/.
- **OASIS XACML1-1**, *eXtensible Access Control Markup Language (XACML) V1.0,* 18 February 2003, http://www.oasis-open.org/apps/org/workgroup/xacml/.
- **OASIS XACML 2.0**, *eXtensible Access Control Markup Language (XACML) V2.0,* 1 February 2005, http://www.oasis-open.org/apps/org/workgroup/xacml/.
- **SSL3**, *The SSL Protocol Version 3.0*. See http://wp.netscape.com/eng/ssl3/draft302.txt.
- **W3C Character Model** (2004), Working draft, 27 October 2005, *Character Model for the World Wide Web 1.0: Normalization.*

## СЕРИИ РЕКОМЕНДАЦИЙ МСЭ-Т

| | |
|---|---|
| Серия A | Организация работы МСЭ-Т |
| Серия D | Общие принципы тарификации |
| Серия E | Общая эксплуатация сети, телефонная служба, функционирование служб и человеческие факторы |
| Серия F | Нетелефонные службы электросвязи |
| Серия G | Системы и среда передачи, цифровые системы и сети |
| Серия H | Аудиовизуальные и мультимедийные системы |
| Серия I | Цифровая сеть с интеграцией служб |
| Серия J | Кабельные сети и передача сигналов телевизионных и звуковых программ и других мультимедийных сигналов |
| Серия K | Защита от помех |
| Серия L | Конструкция, прокладка и защита кабелей и других элементов линейно-кабельных сооружений |
| Серия M | Управление электросвязью, включая СУЭ и техническое обслуживание сетей |
| Серия N | Техническое обслуживание: международные каналы передачи звуковых и телевизионных программ |
| Серия O | Требования к измерительной аппаратуре |
| Серия P | Качество телефонной передачи, телефонные установки, сети местных линий |
| Серия Q | Коммутация и сигнализация |
| Серия R | Телеграфная передача |
| Серия S | Оконечное оборудование для телеграфных служб |
| Серия T | Оконечное оборудование для телематических служб |
| Серия U | Телеграфная коммутация |
| Серия V | Передача данных по телефонной сети |
| **Серия X** | **Сети передачи данных, взаимосвязь открытых систем и безопасность** |
| Серия Y | Глобальная информационная инфраструктура, аспекты межсетевого протокола и сети последующих поколений |
| Серия Z | Языки и общие аспекты программного обеспечения для систем электросвязи |

Отпечатано в Швейцарии
Женева, 2008 г.