Международный союз электросвязи

МСЭ-Т

СЕКТОР СТАНДАРТИЗАЦИИ
ЭЛЕКТРОСВЯЗИ МСЭ

Y.2722

(01/2011)

СЕРИЯ Y: ГЛОБАЛЬНАЯ ИНФОРМАЦИОННАЯ ИНФРАСТРУКТУРА, АСПЕКТЫ ПРОТОКОЛА ИНТЕРНЕТ И СЕТИ ПОСЛЕДУЮЩИХ ПОКОЛЕНИЙ

Сети последующих поколений – Безопасность

Механизмы управления определением идентичности в СПП

Рекомендация МСЭ-Т Y.2722

ITU-T

Международный союз электросвязи

РЕКОМЕНДАЦИИ МСЭ-Т СЕРИИ Y

ГЛОБАЛЬНАЯ ИНФОРМАЦИОННАЯ ИНФРАСТРУКТУРА, АСПЕКТЫ ПРОТОКОЛА ИНТЕРНЕТ И СЕТИ ПОСЛЕДУЮЩИХ ПОКОЛЕНИЙ

ГЛОБАЛЬНАЯ ИНФОРМАЦИОННАЯ ИНФРАСТРУКТУРА	
Общие положения	Y.100–Y.199
Услуги, приложения и промежуточные программные средства	Y.200–Y.299
Сетевые аспекты	Y.300–Y.399
Интерфейсы и протоколы	Y.400–Y.499
Нумерация, адресация и присваивание имен	Y.500–Y.599
Эксплуатация, управление и техническое обслуживание	Y.600–Y.699
Безопасность	Y.700–Y.799
Рабочие характеристики	Y.800–Y.899
АСПЕКТЫ ПРОТОКОЛА ИНТЕРНЕТ	
Общие положения	Y.1000–Y.1099
Услуги и приложения	Y.1100–Y.1199
Архитектура, доступ, возможности сетей и административное управление ресурсами	Y.1200–Y.1299
Транспортирование	Y.1300–Y.1399
Взаимодействие	Y.1400–Y.1499
Качество обслуживания и сетевые показатели качества	Y.1500–Y.1599
Сигнализация	Y.1600–Y.1699
Эксплуатация, управление и техническое обслуживание	Y.1700–Y.1799
Начисление платы	Y.1800–Y.1899
IPTV по СПП	Y.1900–Y.1999
СЕТИ ПОСЛЕДУЮЩИХ ПОКОЛЕНИЙ	
Структура и функциональные модели архитектуры	Y.2000–Y.2099
Качество обслуживания и рабочие характеристики	Y.2100–Y.2199
Аспекты обслуживания: возможности услуг и архитектура услуг	Y.2200–Y.2249
Аспекты обслуживания: взаимодействие услуг и СПП	Y.2250–Y.2299
Нумерация, присваивание имен и адресация	Y.2300–Y.2399
Управление сетью	Y.2400–Y.2499
Архитектура и протоколы сетевого управления	Y.2500–Y.2599
Будущие сети	Y.2600–Y.2699
Безопасность	**Y.2700–Y.2799**
Обобщенная мобильность	Y.2800–Y.2899
Открытая среда операторского класса	Y.2900–Y.2999
Будущие сети	Y.3000–Y.3099

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

Рекомендация МСЭ-Т Y.2722

Механизмы управления определением идентичности в СПП

Резюме

В Рекомендации МСЭ-Т Y.2722 определяются механизмы, которые могут использоваться для удовлетворения требований, связанных с управлением определением идентичности (IdM), и потребностей, обусловленных развертыванием сетей последующих поколений (СПП).

Хронологическая справка

Издание	Рекомендация	Утверждение	Исследовательская комиссия
1.0	МСЭ-Т Y.2722	28.01.2011 г.	13-я

Ключевые слова

Федеративная идентичность, управление определением идентичности, механизмы управления определением идентичности, сеть последующих поколений, безопасность.

ПРЕДИСЛОВИЕ

Международный союз электросвязи (МСЭ) является специализированным учреждением Организации Объединенных Наций в области электросвязи и информационно-коммуникационных технологий (ИКТ). 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 осуществляется на добровольной основе. Однако данная Рекомендация может содержать некоторые обязательные положения (например, для обеспечения функциональной совместимости или возможности применения), и в таком случае соблюдение Рекомендации достигается при выполнении всех указанных положений. Для выражения требований используются слова "следует", "должен" ("shall") или некоторые другие обязывающие выражения, такие как "обязан" ("must"), а также их отрицательные формы. Употребление таких слов не означает, что от какой-либо стороны требуется соблюдение положений данной Рекомендации.

ПРАВА ИНТЕЛЛЕКТУАЛЬНОЙ СОБСТВЕННОСТИ

МСЭ обращает внимание на вероятность того, что практическое применение или выполнение настоящей Рекомендации может включать использование заявленного права интеллектуальной собственности. МСЭ не занимает какую бы то ни было позицию относительно подтверждения, действительности или применимости заявленных прав интеллектуальной собственности, независимо от того, доказываются ли такие права членами МСЭ или другими сторонами, не относящимися к процессу разработки Рекомендации.

На момент утверждения настоящей Рекомендации МСЭ получил извещение об интеллектуальной собственности, защищенной патентами, которые могут потребоваться для выполнения настоящей Рекомендации. Однако те, кто будет применять Рекомендацию, должны иметь в виду, что вышесказанное 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 Y.2722

Механизмы управления определением идентичности в СПП

1 Сфера применения

В [ITU-T Y.2721] "Требования к управлению определением идентичности в СПП, и сценарии использования" описаны требования, связанные с управлением определением идентичности (IdM) в сетях последующих поколений (СПП). В настоящей Рекомендации содержится описание конкретных механизмов IdM и наборов вариантов, которые следует использовать для удовлетворения требований, определенных в [ITU-T Y.2721]. Кроме того, в настоящей Рекомендации содержатся примеры передового опыта и руководящие указания относительно поддержки функциональной совместимости и удовлетворения других потребностей.

Настоящая Рекомендация предназначена для использования совместно с [ITU-T Y.2720] и [ITU-T Y.2721], поскольку основополагающие архитектурные концепции, требования и сценарии использования в ней не повторяются.

ПРИМЕЧАНИЕ. – Пользователи Рекомендации и лица, использующие описанные механизмы, должны соблюдать все применимые национальные и региональные законы, нормативные акты и политические принципы. В некоторых конкретных нормативных и законодательных актах может требоваться реализация механизмов защиты информации, позволяющей установить личность.

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. Перечень действующих на настоящий момент Рекомендаций МСЭ-Т регулярно публикуется. Ссылка на документ в рамках этой Рекомендации не дает ему как отдельному документу статуса Рекомендации.

[ITU-T X.509] Рекомендация МСЭ-Т X.509 (2005 г.) | ISO/IEC 9594-8:2005, *Информационные технологии – Взаимосвязь открытых систем – Справочник: Структуры сертификатов открытых ключей и атрибутов*.

[ITU-T X.1035] Recommendation ITU-T X.1035 (2007), *Password-authenticated key exchange (PAK) protocol*.

[ITU-T X.1141] Рекомендация МСЭ-Т X.1141 (2006 г.), *Язык разметки, предусматривающий защиту данных (SAML 2.0)*.

[ITU-T X.1252] Рекомендация МСЭ-Т X.1252 (2010 г.), *Базовые термины и определения в области управления определением идентичности*.

[ITU-T Y.2012] Recommendation ITU-T Y.2012 (2006), *Functional requirements and architecture of NGN release 1*.

[ITU-T Y.2701] Рекомендация МСЭ-Т Y.2701 (2007 г.), *Требования к безопасности для сетей последующих поколений версии 1*.

[ITU-T Y.2702] Рекомендация МСЭ-Т Y.2702 (2008 г.), *Требования к аутентификации и авторизации для СПП варианта 1*.

[ITU-T Y.2704] Рекомендация МСЭ-Т Y.2704 (2010 г.), *Механизмы и процедуры безопасности для сетей последующих поколений*.

[ITU-T Y.2720] Рекомендация МСЭ-Т Y.2720 (2009 г.), *Структура управления определением идентичности в СПП*.

[ITU-T Y.2721] Recommendation ITU-T Y.2721 (2010), *NGN identity management requirements and use cases*.

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

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

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

3 Определения

В данной Рекомендации используются следующие термины, определенные в [ITU-T Y.2720] и [ITU-T X.1252].

В частности, из [ITU-T X.1252] были взяты следующие определения:

3.1 поставщик данных идентичности (identity provider) (IdP): См. поставщик услуг определения идентичности (IdSP).

3.2 поставщик услуг определения идентичности (identity service provider) (IdSP): Объект, который выполняет верификацию информации об идентичности других объектов, поддерживает эту информацию, управляет ею и может ее создавать и назначать.

4 Сокращения

В настоящей Рекомендации используются следующие сокращения и акронимы:

AKA	Authentication and Key Agreement	Соглашение об аутентификации и ключе
ASP	Application Service Provider	Поставщик прикладных услуг
AuC	Authentication Centre	Центр аутентификации
AV	Authentication Vector	Вектор аутентификации
BSF	Bootstrapping Server Function	Функция начальной загрузки сервера
CK	Ciphering Key	Ключ шифрования
GBA	Generic Bootstrapping Architecture	Общая архитектура начальной загрузки
HSS	Home Subscriber System	Система абонентских данных
IdM	Identity Management	Управление определением идентичности
IdP	Identity Provider	Поставщик данных идентичности
IdSP	Identity Service Provider	Поставщик услуг определения идентичности
IK	Integrity Key	Ключ целостности
IMPI	IP Multimedia Private user Identity	Закрытый идентификатор абонента системы передачи мультимедийных данных на базе протокола Интернет
IMPU	IP Multimedia Public User identity	Открытый идентификатор абонента системы передачи мультимедийных данных на базе протокола Интернет
IMS	IP Multimedia Subsystem	Подсистема передачи мультимедийных данных на базе протокола Интернет
IMSI	International Mobile Subscriber Identity	Международный идентификатор абонента подвижной связи
IPTV	Internet Protocol Television	Телевидение по протоколу Интернет
ISIM	IMS Subscriber Identity Module	Модуль идентичности абонента IMS
LDAP	Lightweight Directory Access Protocol	Облегченный протокол доступа к сетевому каталогу
MS	Mobile Station	ПС Подвижная станция
NAF	Network Application Function	Сетевая прикладная функция
NGN	Next Generation Networks	СПП Сети последующих поколений
OASIS	Organization for the Advancement of Structured Information Standards	Организация по развитию стандартов структурированной информации

OTP	One Time Password	Однократный пароль
PII	Personally Identifiable Information	Информация, позволяющая установить личность
PKI	Public Key Infrastructure	Инфраструктура открытых ключей
RP	Relying Party	Полагающаяся сторона
SAML	Security Assertion Markup Language	Язык разметки утверждений безопасности
SIP	Session Initiation Protocol	Протокол инициирования сеанса связи
SLF	Subscriber Locator Function	Функция указателя абонента
SOAP	Simple Object Access Protocol	Простой протокол доступа к объектам
SQL	Structured Query Language	Язык структурированных запросов
SSO	Single Sign-On	Однократная регистрация
UE	User Equipment	Оборудование пользователя
UICC	Universal Integrated Circuit Card	Универсальная смарт-карта
UMTS	Universal Mobile Telecommunications System	Универсальная система подвижной электросвязи
USIM	Universal Subscriber Identifier Module	Универсальный модуль идентификатора абонента
WAP	Wireless Application Protocol	Протокол беспроводных приложений
WSS	Web Services Security	Безопасность веб-услуг
XML	eXtensible Markup Language	Расширяемый язык разметки
XRDS	eXtensible Resource Descriptor Sequence	Расширяемая последовательность дескрипторов ресурса

5 Соглашения по терминологии

В настоящей Рекомендации:

Ключевые слова "требуется, чтобы" обозначают требование, которое должно строго соблюдаться, и от которого не допускается отклонений, если должно быть заявлено соответствие данной Рекомендации.

Ключевые слова "рекомендуется" обозначают требование, которое рекомендовано, но не обязательно требуется. Поэтому для заявления о соответствии это требование не обязательно.

Ключевые слова "запрещено, чтобы" обозначают требование, которое должно строго соблюдаться, и от которого не допускается отклонений, если должно быть заявлено соответствие данной Рекомендации.

Ключевые слова "необязательно можно" обозначают необязательное требование, которое допускается, не предполагая никакого рекомендательного оттенка. Этот термин не предназначен для утверждения, что реализация поставщика должна обеспечивать этот вариант, а функцию может необязательно предоставлять оператор сети/поставщик услуг. Наоборот, это значит, что поставщик может необязательно предоставлять эту функцию и тем не менее заявлять о соответствии техническим условиям данной Рекомендации.

В тексте данной Рекомендации и ее дополнениях иногда встречаются слова *должен*, *не должен*, *следует* и *может*, в этом случае их следует понимать как *требуется, чтобы*; *запрещено, чтобы*; *рекомендуется* и *дополнительно можно*, соответственно. Появление таких фраз или ключевых слов в дополнении или материалах, однозначно помеченных, как *информативных*, должно пониматься, как не несущее нормативного смысла.

6 Механизмы и процедуры, поддерживающие функции IdM

6.1 Управление жизненным циклом идентичности

Информацию об управлении жизненным циклом идентичности см. в [ITU-T Y.2720] "Структура управления определением идентичности в СПП".

6.2 Аутентификация и гарантия аутентификации

В данном пункте описываются механизмы аутентификации и гарантии идентичности и информации, подтверждающей идентичность. В разделе делается ссылка на механизмы аутентификации, описанные в других документах.

В зависимости от контекста и необходимого уровня гарантии в конкретных приложениях или услугах IdSP могут использовать конкретные механизмы аутентификации, такие как аутентификация на базе веб-услуг (WS), профиля языка разметки утверждений безопасности (SAML), аутентификация по сертификатам или аутентификация по паролю (включая однократные пароли (OTP)). Метод (или методы) аутентификации выбираются исходя из требований к уровню гарантии. IdSP может запрашивать информацию в целях определения методов аутентификации, удовлетворяющих требованиям к уровню гарантии поставщика услуги.

6.2.1 Аутентификация по профилю SAML безопасности веб-услуг

6.2.1.1 Утверждения SAML

Язык разметки утверждений безопасности (SAML) [ITU-T X.1141] определяет формат утверждений, которые могут использоваться в IdM для обмена информацией о безопасности. К функциям IdM, которые возможно реализовать с использованием SAML, относятся: аутентификация, совместное использование атрибутов и авторизация, соответствующие трем типам заявлений относительно субъекта утверждения SAML:

- Заявление об аутентификации – передает информацию о том, что субъект утверждения, был аутентифицирован с помощью определенных средств и в определенное время.
- Заявление об атрибутах – передает информацию о том, что субъект утверждения, связан с перечисленными атрибутами.
- Заявление о решении по результатам аутентификации – передает информацию о том, что субъекту утверждения был предоставлен доступ к определенным ресурсам или что ему было отказано в таком доступе.

Содержание утверждения SAML на высоком уровне может быть описано следующим образом: утверждение **A** было выпущено в момент **t** отправителем **R** относительно субъекта **S** при соблюдении условий **C**.

Утверждения SAML, используемые для сообщения информации об аутентификации, атрибутах и авторизации, передаются в сообщениях простого протокола доступа к объектам (SOAP). При обмене сообщениями SOAP по незащищенному транспортному средству настоятельно рекомендуется использовать подпись XML [b-W3C XML signature] для проверки отношений между сообщением SOAP и заявлениями об утверждениях, переносимых в этом сообщении. Стандарт "Безопасность веб-услуг: Профиль жетонов SAML" (*Web Services Security (WSS): SAML Token Profile)* [b-OASIS SAML token] описывает как:

- Утверждения SAML (также называемые жетонами SAML) переносятся в сообщении SOAP и на них делается ссылка из сообщений SOAP.
- Подпись XML используется для связывания субъекта и заявлений утверждения SAML с сообщением SOAP.

Типовое использование жетона SAML с сообщением SOAP, сформированным в соответствии с настоящей Рекомендацией, показано на рисунке 1 и описано ниже.

В этом примере подписанное сообщение SOAP содержит утверждение SAML с заявлением об атрибутах. На основании содержащейся в этом заявлении информации приемник может принимать решения об управлении доступом.

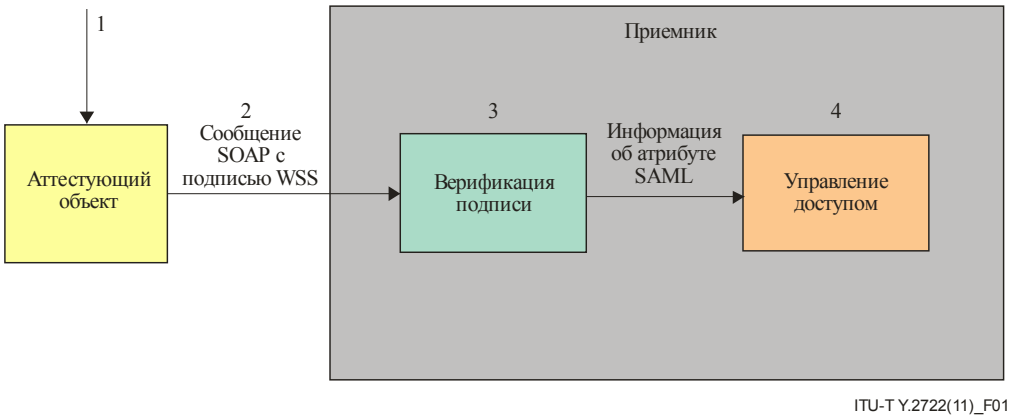


Рисунок 1 – Типовые шаги формирования и обработки сообщения SOAP с жетоном SAML

1) Аттестующий объект получает утверждение SAML с заявлением об атрибутах и формирует и включает его в сообщение SOAP согласно [b-OASIS SAML token].

2) Аттестующий объект отправляет сообщение SOAP с подписью WSS приемнику.

3) Приемник осуществляет верификацию цифровой подписи.

4) Информация заявления SAML может использоваться для принятия решений об управлении доступом.

6.2.1.2 Методы подтверждения субъекта жетонов SAML

Стандарт OASIS "Безопасность веб-услуг: Профиль жетонов SAML 1.1" (*Web Services Security: SAML Token Profile 1.1)* [b-OASIS SAML token] описывает, как присоединить утверждение SAML к сообщению SOAP, а также определяет два обязательных метода подтверждения субъекта:

- держатель ключа;
- отправитель ручается.

Основные элементы XML сообщения SOAP, сформированного согласно [b-OASIS WSS SOAP], представлены на рисунке 2.

Утверждение SAML помещается в заголовок <wsse:Security>, который также содержит цифровую подпись <ds:Signature>. Цифровая подпись используется приемником сообщения SOAP для проверки того, что отправитель сообщения знает ключ, используемый для расчета подписи по дайджесту тела SOAP, и для контроля целостности. Алгоритмом формирования дайджеста является SHA 1, а алгоритмом формирования подписи – RSA_SHA 1, определенный в [b-OASIS WSS SOAP]. Значение подписи предоставляется в элементе <ds:SignatureValue> цифровой подписи <ds:Signature>.

В двух методах подтверждения субъекта определены разные способы передачи приемнику информации о ключе.

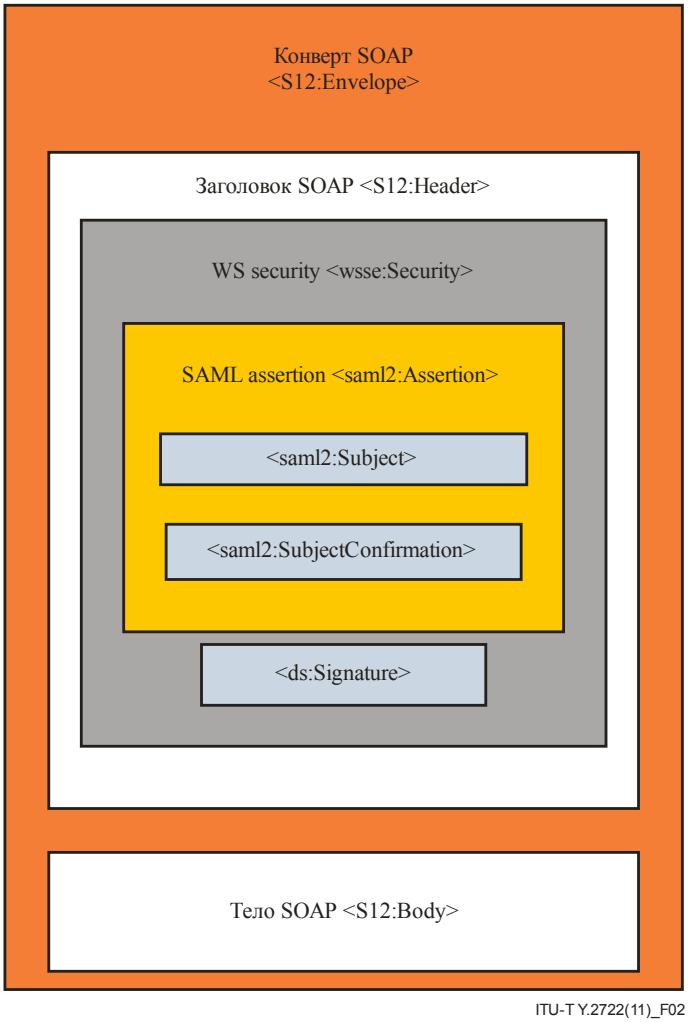


Рисунок 2 – Структура сообщения SOAP с утверждением SAML

В нижеследующих пунктах описаны два метода подтверждения субъекта.

6.2.1.2.1 Метод подтверждения субъекта "держатель ключа"

На рисунке 3 показана структура утверждения SAML, используемого для метода подтверждения субъекта как держателя ключа. Атрибут метода элемента <saml2:SubjectConfirmation> показывает метод подтверждения субъекта (держатель ключа).

Этот метод предусматривает, что отправитель (также называемый аттестующий объект) должен доказать свою правомочность делать заявления относительно субъекта, продемонстрировав знание ключа, который определяется в элементе <ds:KeyValue>, содержащемся в элементе <ds:KeyInfo> утверждения SAML. Элемент <ds:KeyInfo> указывает открытый или закрытый ключ, который используется для подтверждения идентичности субъекта. Далее этот метод определяет, что отправитель может сделать это, подписав дайджест тела SOAP, используя этот ключ. Подпись содержится в элементе <ds:Signature> заголовка безопасности WS, как показано на рисунке 2.

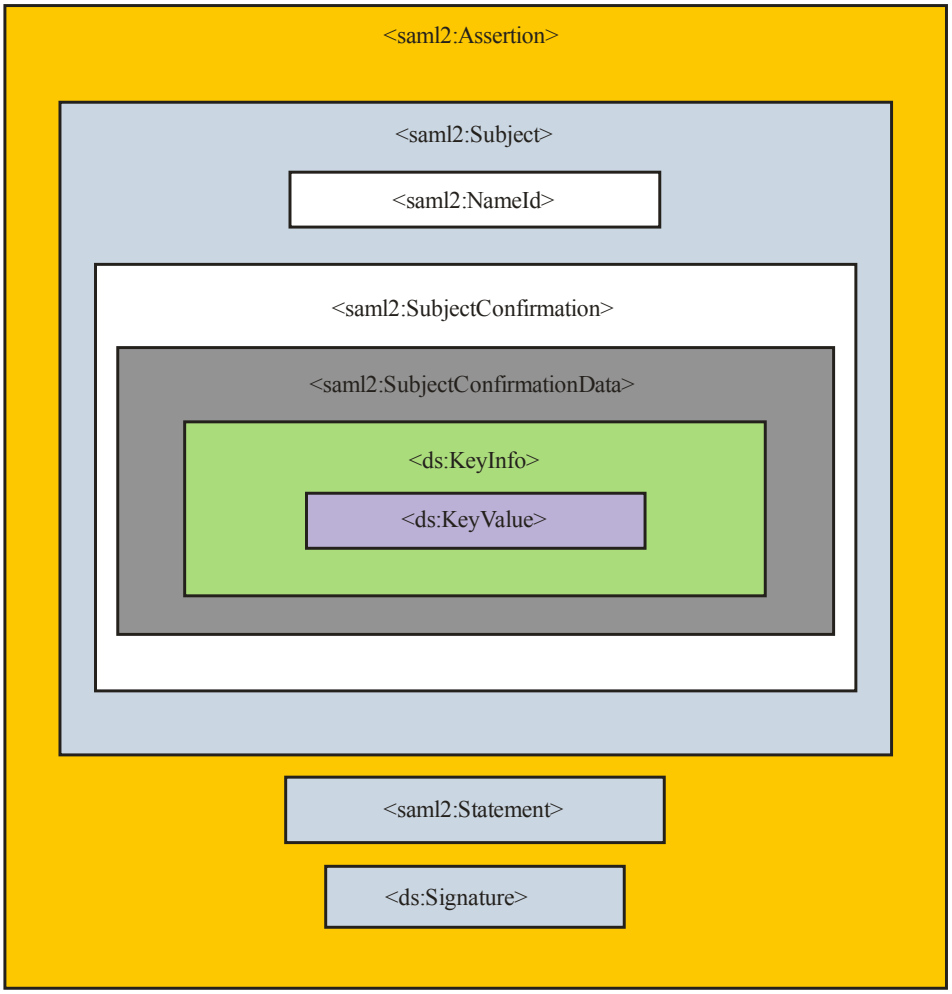


Рисунок 3 – Структура утверждения SAML, используемого для метода подтверждения субъекта как держателя ключа

Приемник сообщения SOAP получает ключ, используя информацию, представленную аттестующим объектом в элементе <ds:KeyInfo>. Далее приемник рассчитывает цифровую подпись тела SOAP и проверяет, совпадает ли она с подписью, представленной аттестующим объектом. Если совпадает, то субъект и заявления утверждения SAML могут быть отнесены к аттестующему объекту, а содержимое тела SOAP, целостность которого защищается ключом, может рассматриваться как представленное аттестующим объектом.

6.2.1.2.2 Метод подтверждения субъекта "отправитель ручается"

На рисунке 4 показана структура утверждения SAML, используемого для метода подтверждения субъекта "отправитель ручается". Атрибут метода элемента <saml2:SubjectConfirmation> указывает метод подтверждения субъекта (отправитель ручается).

Аттестующий объект уполномочивается приемником делать утверждение SAML относительно субъекта при условии, что значение атрибута метода элемента <SubjectConfirmation> указывает метод "отправитель ручается".

Аттестующий объект получает одно или несколько утверждений или ссылок на утверждения от одного или нескольких органов и включает их в сообщение SOAP. Далее он рассчитывает подпись дайджеста утверждений SAML и тела сообщения SOAP. Подпись содержится в элементе <ds:Signature> заголовка безопасности веб-услуг (который показан на рисунке 2). Аттестующий объект необязательно предоставляет приемнику информацию о ключе, который использовался для

расчета подписи. Если такая информация отсутствует, предполагается, что приемник определяет ключ иными способами.

Приемник выполняет валидацию подписи. Если подпись действительна, приемник устанавливает факт того, что заявления были сделаны аттестующим объектом о субъекте.

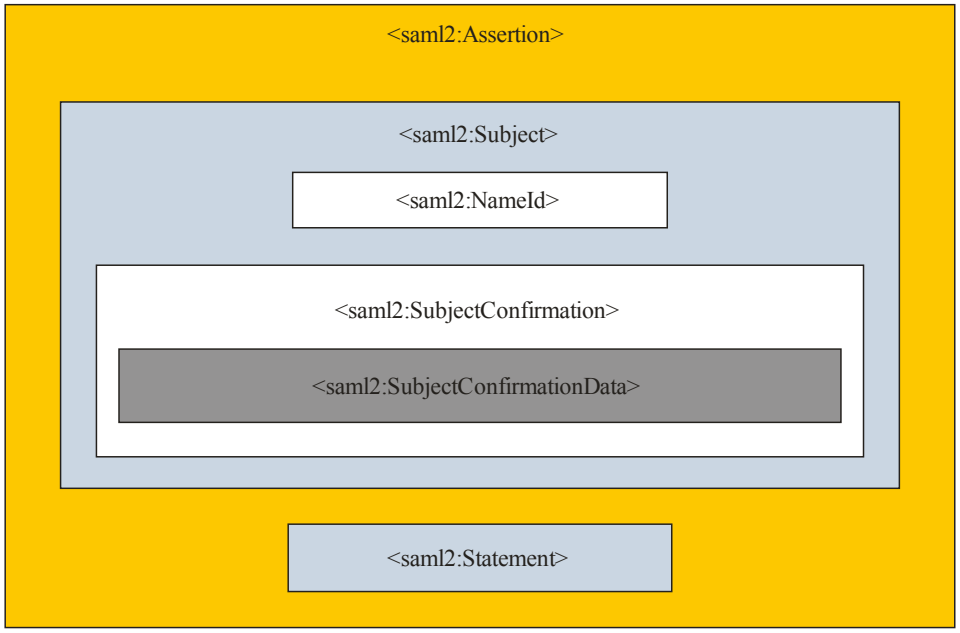


Рисунок 4 – Структура утверждения SAML, которое используется для метода подтверждения субъекта "получатель ручается"

6.2.2 Аутентификация по сертификату

В зависимости от контекста и необходимого уровня гарантии в конкретном приложении или услуге могут использоваться сертификаты [ITU-T X.509]. Использование сертификатов [ITU-T X.509] для аутентификации описано в [ITU-T Y.2704] "Механизмы и процедуры безопасности для сетей последующих поколений".

6.2.3 Аутентификация по паролю

В зависимости от контекста и необходимого уровня гарантии в конкретном приложении или услуге может использоваться аутентификация по паролю. Описание механизма аутентификации по паролю содержится в [ITU-T X.1035].

6.2.4 Однократный пароль

В зависимости от контекста и необходимого уровня гарантии в конкретном приложении или услуге может использоваться однократный пароль (OTP). Один из методов реализации OTP описан в [IETF RFC 2289].

6.2.5 Использование соглашения об аутентификации и ключе (AKA) для взаимной аутентификации

Протокол соглашения об аутентификации и ключе (AKA) универсальной системы подвижной электросвязи (UMTS) может использоваться для обеспечения взаимной аутентификации подвижной станции (ПС) и сети. UMTS AKA – это запросно-ответный протокол, в котором происходит долговременное совместное использование ключей универсальным модулем идентичности абонента (USIM) и центром аутентификации (AuC). Эти объекты размещаются в универсальной смарт-карте (UICC) ПС и домашней сети ПС, соответственно. В определенных коммерческих соглашениях функции AuC может обеспечивать IdSP. Протокол AKA описан в [ATIS 33102].

6.2.6 Объединение аутентификации на базе PKI с IMS

Безопасность подсистем передачи мультимедийных данных на базе протокола Интернет (IMS) базируется на механизме AKA, в котором используется разделенный секрет и запросно-ответный протокол для аутентификации пользователь-сеть. Вместе с тем безопасность определенных услуг СПП (например, IPTV) базируется на сертификатах инфраструктуры открытых ключей (PKI). Для обеспечения возможности стыковки услуг СПП, использующих сертификаты PKI, с безопасностью IMS, может быть целесообразно объединить аутентификацию на базе PKI с аутентификацией IMS таким образом, чтобы использовать преимущества безопасности IMS.

Объединение аутентификации IMS с аутентификацией на базе PKI позволяет пользовательскому оборудованию сети осуществлять взаимную аутентификацию на основе своих соответствующих сертификатов и согласовывать набор шифроключей на основе того же поколения ключей, что и используемое в AKA. Для этого оборудование пользователя и сеть должны быть обеспечены соответствующими персональными ключами и сертификатами и иметь возможность выполнять операции PKI.

В отношении соглашения о шифроключе (*CK*) и ключе целостности (*IK*) предпочтительный механизм объединения определяет два варианта:

1) достижение соглашения по ключам *CK* и *IK* с использованием секрета, разделенного между функцией конечного пользователя и функциональным объектом профиля пользователя S-5 (SUP-FE), который определен в [ITU-T Y.2012];

2) достижение соглашения о ключах *CK* и *IK* без использования разделенного секрета.

Общая последовательность обработки вызова в первом и втором вариантах представлена, соответственно, на рисунках 5 и 6.

6.2.6.1 Условные обозначения

В данном пункте используются следующие условные обозначения:

– "|" обозначает конкатенацию строк;

– *CK* обозначает шифроключ;

– *IK* обозначает ключ целостности;

– *K()* обозначает шифрование симметричным ключом;

– N_{pr} *[]* обозначает шифрование с помощью сетевого личного ключа N_{pr};

– N_{pu} *[]* обозначает шифрование с помощью сетевого открытого ключа N_{pu}, доступного в сертификате сети;

– U_{pr} *[]* обозначает шифрование с помощью личного ключа пользователя U_{pr}.

6.2.6.2 Объекты, участвующие в аутентификации

- S-5 – функциональный объект профиля пользователя услуги (SUP-FE).
- Функция конечного пользователя. Этот объект может запускать приложение клиента SIP.
- Функциональный объект управления сеансами связи S-n (CSC-FE), где S-n означает один из следующих объектов:
  - S-1 функциональный объект обслуживающей функции управления сеансами связи (S-CSC-FE);
  - S-2 функциональный элемент управления сеансами связи с прокси-элементом (P-CSC-FE);
  - S-3 функциональный элемент запрашивающей функции управления сеансами связи (I-CSC-FE).

S-n используется для обозначения одного из таких объектов, когда между ними нет различия в контексте описываемой процедуры аутентификации. Описания функциональных объектов СПП (S-1, S-2, S-3, S-5 и функции конечного пользователя) содержатся в [ITU-T Y.2012].

6.2.6.3 Достижение соглашения относительно ключей *CK* и *IK* с использованием секрета, разделенного между функцией конечного пользователя и S-5 (вариант 1)

Процедура обработки вызова представлена на рисунке 5. Ниже описаны основные шаги:

1) Функция конечного пользователя направляет S-n запрос "Выполнить регистрацию SIP" с *IMPU* и *IMPI* пользователя.

2) S-1 запрашивает случайный вызов *RAND*, *CK* и *IK* от S-5. Значения *RAND*, *CK* и *IK* определены в [ATIS 33102].

3) S-1 принимает *RAND*, *CK* и *IK* от S-5 для пользователя.

4) S-n направляет функции конечного пользователя сообщение "SIP 401 не авторизован" с вызовом *RAND* и его зашифрованным значением $N_{pr}[RAND]$.

   Функция конечного пользователя:

   - принимает значения *A*, которое предположительно равно *RAND*, и значения *B*, которое предположительно равно $N_{pr}[RAND]$;
   - получает сетевой открытый ключ N_{pu};
   - дешифрует *B* с помощью N_{pu} и сравнивает результат с *A*. Если значения равны между собой, значит сеть авторизована, если не равны – процедура аутентификации прерывается;
   - генерирует *IK* и *CK*, используя разделенный секрет K_s;
   - генерирует значение $U_{pr}[N_{pu}\ [K]|K(RAND)]$.

5) Функция конечного пользователя направляет S-n сообщение "Выполнить регистрацию SIP" вместе с идентификаторами *IMPU* и *IMPI* и значение $U_{pr}[N_{pu}[K]|K(RAND)]$.

6) S-1 направляет S-5 данные, полученные на шаге 5, и запрашивает верификацию записи пользователя.

   S-5:

   - запрашивает сертификат пользователя для получения открытого ключа пользователя U_{pu};
   - дешифрует с помощью U_{pu} полученное значение *C*, которое предположительно равно $U_{pr}[N_{pu}[K]|K(RAND)]$, для извлечения значения *D*|*E*, где *D* предположительно равно $N_{pu}[K,]$ а *E* предположительно равно *K(RAND)*;
   - дешифрует с помощью сетевого личного ключа N_{pr} значение *D* для получения *K′*;
   - дешифрует с помощью *K′* значение *E* для получения *RAND′*;
   - сравнивает *RAND′* с *RAND*. Их совпадение означает, что пользователь прошел аутентификацию.

7) S-5 сообщает S-1 результат аутентификации и запись пользователя.

8) S-1 использует эту запись для проверки, разрешено ли аутентифицированному пользователю регистрироваться и получать запрошенную услугу. Если это так, S-n извещает функцию конечного пользователя о том, что доступ предоставлен, используя сообщение "SIP 200 OK".

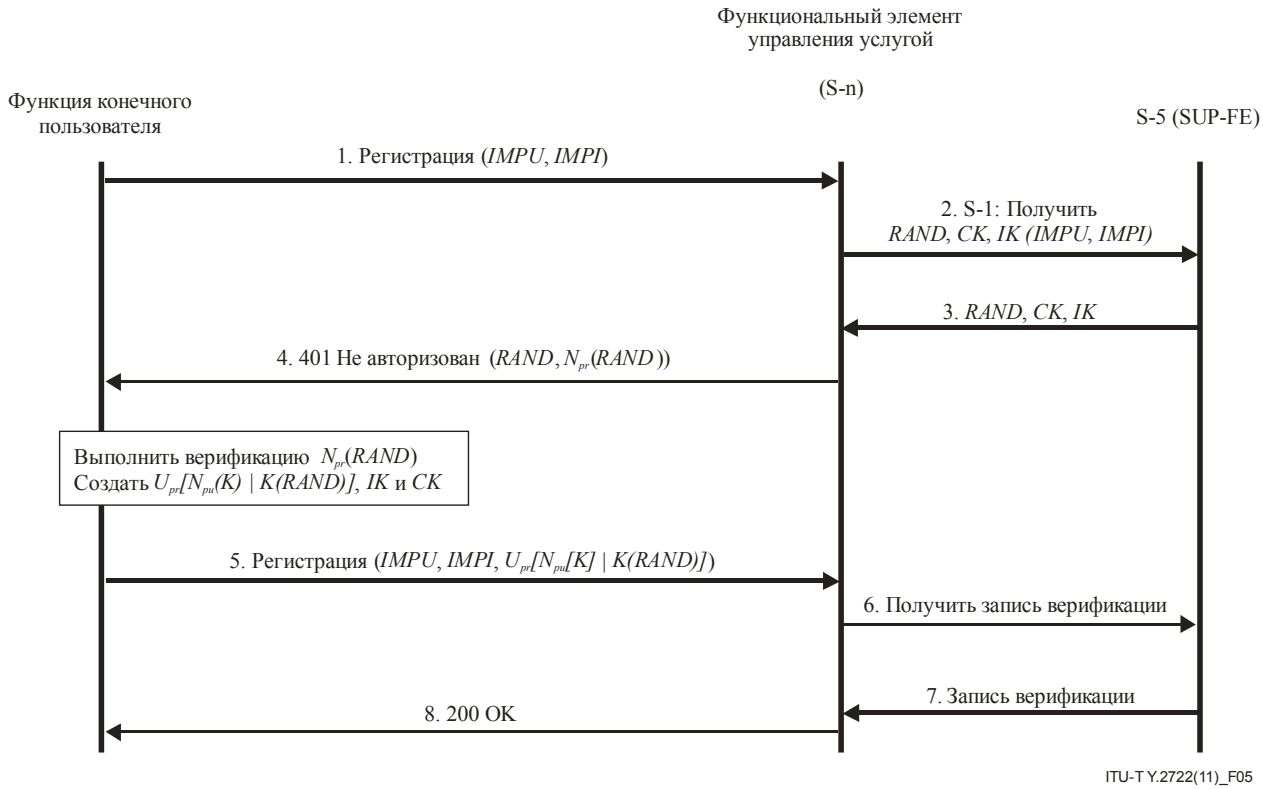


Рисунок 5 – Объединение механизма аутентификации IMS с аутентификацией на базе PKI (вариант 1)

6.2.6.4 Достижение соглашения относительно ключей CK и IK без использования секрета, разделенного между функцией конечного пользователя и S-5 (вариант 2)

Процедура обработки вызова представлена на рисунке 6. Ниже описаны основные шаги:

1) Функция конечного пользователя направляет S-n запрос "Выполнить регистрацию SIP" с *IMPU* и *IMPI* пользователя.

2) S-1 запрашивает случайный вызов *RAND* от S-5. Значение *RAND* определено в [ATIS 33102].

3) S-1 принимает *RAND* от S-5 для конкретного пользователя.

4) S-n направляет функции конечного пользователя сообщение "SIP 401 не авторизован" с вызовом *RAND* и его зашифрованным значением $N_{pr}[RAND]$.

   Функция конечного пользователя:

   - принимает значения *A*, которое предположительно равно *RAND*, и значения *B*, которое предположительно равно $N_{pr}[RAND]$;
   - получает сетевой открытый ключ N_{pu};
   - дешифрует *B* с помощью N_{pu} и сравнивает результат с *A*. Если значения равны между собой, значит сеть авторизована, если не равны – процедура аутентификации прерывается;
   - генерирует *IK* и *CK*, используя случайно генерируемый ключ *K*;
   - генерирует значение $U_{pr}[N_{pu}[K]|K(RAND)]$.

5) Функция конечного пользователя направляет S-n сообщение "Выполнить регистрацию SIP" вместе с идентификаторами *IMPU* и *IMPI* и значение $U_{pr}[N_{pu}[K]|K(RAND)]$.

6) S-1 направляет S-5 данные, полученные на шаге 5, и запрашивает верификацию записи пользователя и ключи *CK* и *IK*.

S-5:

- запрашивает сертификат пользователя для получения открытого ключа пользователя U_{pu};
- дешифрует с помощью U_{pu} полученное значение *C*, которое предположительно равно $U_{pr}[N_{pu}[K]|K(RAND)]$, для извлечения значения *D*|*E*, где *D* предположительно равно $N_{pu}[K]$, а *E* предположительно равно *K(RAND)*;
- дешифрует с помощью сетевого личного ключа N_{pr} значение *D* для получения *K′*;
- дешифрует с помощью *K′* значение *E* для получения *RAND′*;
- сравнивает *RAND′* с *RAND*. Их совпадение означает, что пользователь прошел аутентификацию и *K′* = *K*. То есть функция конечного пользователя и S-5 теперь совместно используют ключ *K*;
- генерирует ключи *CK* и *IK*, используя совместный ключ *K*. Например, используя ключ *K* в качестве исходного параметра, для генерации *CK* и *IK* могут использоваться одни и те же функции, определенные в [ATIS 33102].

7) S-5 сообщает S-1 результат аутентификации, запись пользователя и ключи *CK* и *IK*.

8) S-1 использует эту запись для проверки, разрешено ли аутентифицированному пользователю регистрироваться и получать запрошенную услугу. Если это так, S-n извещает функцию конечного пользователя о том, что доступ предоставлен, используя сообщение "SIP 200 OK".

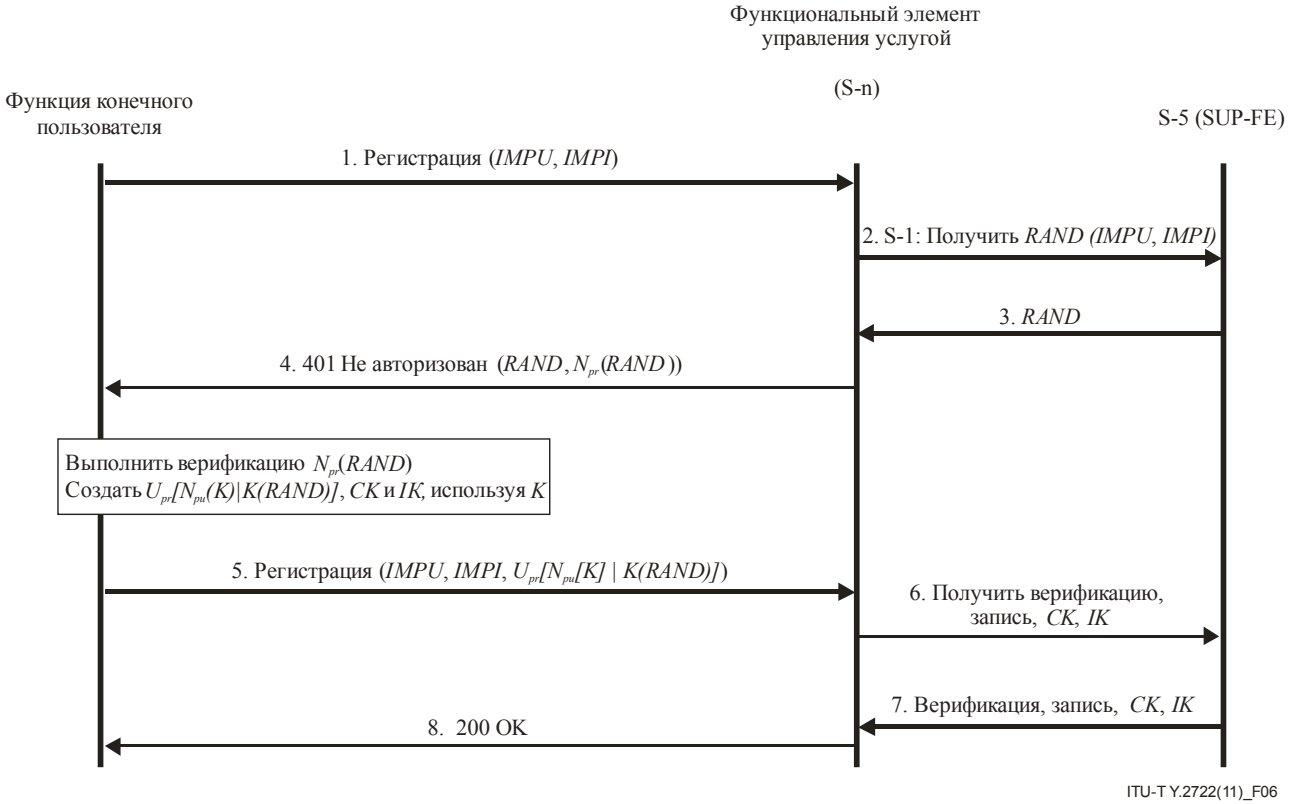


Рисунок 6 – Объединение механизма аутентификации IMS с аутентификацией на базе PKI (вариант 2)

6.2.6.5 Сравнение варианта 1 и варианта 2

В таблице 1 проведено сравнение механизмов, описанных в вариантах 1 и 2.

Таблица 1 – Сравнение варианта 1 и варианта 2 достижения соглашения о ключе между функцией конечного пользователя и S-5 относительно ключей *CK* и *IK*

	Вариант 1 (с предварительно разделенным секретом)	**Вариант 2 (без предварительно разделенного секрета)**
Преимущества	Полностью повторное использование механизма AKA для достижения соглашения о ключах *CK* и *IK*	Не требует предоставления секрета, разделенного между функцией конечного пользователя и S-5
Недостатки	Требует предоставления секрета, разделенного между функцией конечного пользователя и S-5	Требует внесения изменений в приложения, запускаемые в функции конечного пользователя (например, в смарт-карте) и S-5, для того чтобы разрешить использование соглашения о ключах *CK* и *IK*

Вариант 1 следует выбирать для упрощения соглашения о ключе применительно к ключам *CK* и *IK*, если функция конечного пользователя и S-5 совместно используют секрет. Вариант 2 следует выбирать, если функция конечного пользователя и S-5 не имеют разделенного между ними секрета.

Реализация этого механизма объединения должна поддерживать оба варианта.

Требования к функциональному объекту S-5

S-5 должен обладать следующими возможностями в дополнение к определенным в [ATIS 33102]:

- хранение сертификатов пользователя и сети в хранилище сертификатов и получение этих сертификатов из хранилища;
- выполнение дешифрования на базе PKI, которое описано на шаге 6 (для обоих вариантов);
- выполнение протокола Diameter, измененного для передачи информации, описанной на шаге 6 (для обоих вариантов), и информации, необходимой для согласования с функцией конечного пользователя аутентификации на базе PKI;
- достижение с функцией конечного пользователя соглашения о методе аутентификации на базе PKI.

6.2.6.6 Требования к функции конечного пользователя

Функция конечного пользователя должна обладать возможностью выполнения следующих функций:

- безопасное хранение личного ключа пользователя U_{pr};
- безопасное хранение секрета K_s, разделенного с сетью (только для варианта 1);
- хранение сертификата сети МСЭ-Т X.509 вместе с открытым ключом сети N_{pu};
- генерирование случайным образом однократного сеансового ключа K и выполнение симметричного шифрования с помощью ключа K;
- генерирование ключей *CK* и *IK* с использованием разделенного секрета K_s согласно [ATIS 33102] (только для варианта 1);
- генерирование ключей *CK* и *IK* согласно описанию шага 6 для варианта 2;
- выполнение шифрования и дешифрования на базе PKI согласно описанию шагов 4 и 5 для обоих вариантов;
- выполнение приложения клиента SIP с модифицированным протоколом SIP, позволяющее передачу информации, описанной на шагах 4 и 5;
- достижение с S-2 соглашения об использовании аутентификации на базе PKI.

6.2.6.7 Требования к S-1

В отношении S-1 действуют дополнительные требования, согласно которым он должен обладать следующими возможностями:

- формировать сообщения SIP с информацией, описанной на шаге 4 (для обоих вариантов);
- извлекать из сообщений SIP информацию, описанную на шаге 5, и распаковывать ее в сообщения Diameter, как описано на шаге 6 (для обоих вариантов);
- выполнять шифрование на базе PKI, описанное на шаге 4 (для обоих вариантов);
- понимать поступающее от S-5 извещение об использовании аутентификации на базе PKI.

6.2.6.8 Требования к интерфейсам SIP между участвующими объектами

Функция конечного пользователя и S-1 осуществляют связь через функциональные объекты S-2 и S-3. Объекты S-2 и S-3 не являются важными в контексте описываемого метода аутентификации и не показаны на рисунках 5 и 6.

Интерфейсы SIP существуют между:

- функцией конечного пользователя и S-2;
- S-2 и S-3;
- S-1 и S-3.

Эти интерфейсы должны обладать возможностью согласования применения аутентификации на базе PKI (включая конкретный вариант для генерации ключа) и передачи информации, описанной на шагах 4 и 5 (для обоих вариантов).

6.2.6.9 Требования к интерфейсам Diameter между участвующими объектами

Интерфейсы Diameter существуют между:

- S-1 и S-5;
- S-3 и S-5.

Эти интерфейсы должны обладать возможностью согласования применения аутентификации на базе PKI (включая конкретный вариант для генерации ключа) и передачи информации, описанной на шаге 6 (для обоих вариантов).

6.2.7 Объединение аутентификации на базе PKI и механизмов утверждений SAML

SAML допускает наличие одного объекта (например, IdSP) для выполнения аутентификации и другого объекта (полагающаяся сторона, например, поставщик прикладных услуг) для использования результатов аутентификации. При таком сценарии IdSP может реализовать несколько методов аутентификации, когда поставщик прикладных услуг (ASP) полагается на утверждения SAML поставщика IdSP. Этот сценарий обеспечивает преимущества и поставщикам IdSP, и поставщикам ASP. Поставщики ASP получают следующие преимущества:

- ASP не должны реализовывать большое число методов аутентификации;
- ASP может поддерживать широкий выбор прикладных услуг с различными требованиями к гарантии аутентификации.

Поставщики IdSP получают следующие преимущества:

- IdSP может предложить услуги IdM, в частности аутентификацию, многим поставщикам ASP;
- IdSP (особенно если IdSP является поставщиком СПП) может использовать свою развернутую инфраструктуру аутентификации, для того чтобы предлагать услуги IdM другим поставщикам.

В данном пункте описывается механизм аутентификации клиента с применением утверждений SAML и аутентификации на базе PKI. Этот механизм вместе с механизмом, описанным в пункте 6.2.6 "Объединение аутентификации на базе PKI с IMS", дает возможность поставщикам СПП максимально эффективно использовать свою инфраструктуру на базе PKI.

В основу этого механизма положена связь перенаправления HTTP SAML, определение корой содержится в [ITU-T X.1141].

6.2.7.1 Объекты, участвующие в аутентификации, и информационный поток

- Функция конечного пользователя. Этот объект может запускать приложение веб-клиента и поддерживать аутентификацию на базе PKI [ITU-T X.509].
- Сервер приложений (AS) – объект, предоставляющий веб-услугу. Он играет роль полагающейся стороны и действует как запрашивающий элемент SAML, описание которого содержится в [ITU-T X.1141].
- A-2 – функциональный объект шлюза приложений (APL-GW-FE), который позволяет выполнять аутентификацию на базе PKI и действует как отвечающий элемент, описание которого содержится в [ITU-T X.1141].
- S-5 – функциональный объект профиля пользователя услуги (SUP-FE).

Информационный поток процедуры аутентификации показан на рисунке 7. Основные шаги обмена данными для аутентификации на базе PKI с утверждением SAML описаны ниже. Описания функциональных объектов СПП (функции конечного пользователя, AS, A-2 и S-5) содержатся в [ITU-T Y.2012].

6.2.7.2 Соглашения по терминологии

В описании используются следующие соглашения по терминологии:

"|" обозначает конкатенацию строк;

K() обозначает шифрование симметричным ключом;

K_s обозначает секрет, разделенный между A-2 и AS;

N_{pr} *[]* обозначает шифрование с помощью сетевого личного ключа N_{pr};

N_{pu} *[]* обозначает шифрование с помощью сетевого открытого ключа N_{pu}, доступного в сертификате сети;

U_{pr} *[]* обозначает шифрование с помощью личного ключа пользователя U_{pr};

RAND обозначает случайно генерируемый запрос.

6.2.7.3 Параметры механизма

В данном пункте описаны параметры, определяемые механизмом. Ниже приводится список параметров:

`pki-auth-challenge` – параметр для передачи значения *RAND*;

`pki-auth-challenge-encrypted` – параметр для передачи значения N_{pr}*[RAND]*;

`pki-auth-user-signature` – параметр для передачи значения $U_{pr}[N_{pu}[K]|K(RAND)]$;

`pki-auth-keyinfo` – параметр для передачи значения $K_s(K)$.

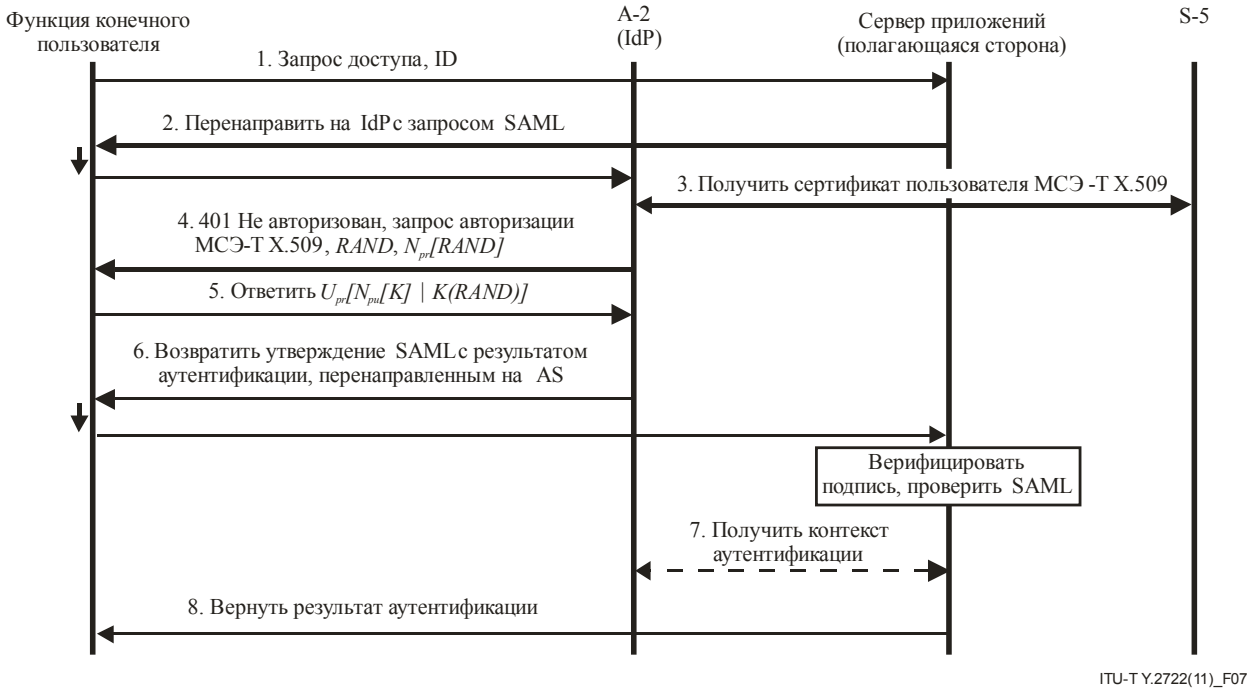


Рисунок 7 – Основные шаги обмена данными для аутентификации на базе PKI с утверждением SAML

Взаимная аутентификация функции конечного пользователя и объекта A-2 аналогична процедуре, используемой механизмом объединения аутентификации на базе PKI с аутентификацией IMS, которая описана в пункте 6.2.6.

Ниже описаны основные шаги процедуры, которая опирается на аутентификацию на базе PKI и утверждения SAML:

1) Веб-клиент функции конечного пользователя выпускает запрос доступа HTTP к серверу приложений (AS). Этот запрос включает идентификатор пользователя и URL объекта A-2.

2) Сервер приложений, действуя как запрашивающий элемент SAML, отвечает на запрос HTTP, направляя запрос SAML. Запрос SAML шифруется в заголовок местоположения запроса HTTP, статус HTTP которого установлен равным 302 или 303. Агент функции конечного пользователя доставляет запрос SAML путем выпуска запроса HTTP ПОЛУЧИТЬ к A-2, который действует как отвечающий элемент SAML. Эта процедура перенаправления HTTP, известная как связь перенаправления HTTP, описана в [ITU-T X.1141]. Для обеспечения аутентификации и целостности зашифрованного в URL сообщения оно должно быть подписано, как указано в пункте 10.2.4.5.2 "Аспекты безопасности" [ITU-T X.1141]. Для подписания следует использовать разделенный секрет K_s.

3) После валидации подписи A-2 получает от S-5 сертификат конечного пользователя и проверяет, действителен ли этот сертификат. Сертификат содержит открытый ключ функции конечного пользователя.

4) A-2 отвечает функции конечного пользователя сообщением HTTP "ответить", указывая, что требуется аутентификация с использованием сертификата [ITU-T X.509]. Это сопровождается установлением значения заголовка ответа `WWW-Authenticate` [b-IETF RFC 2616] в "`pki-auth`". Тело сообщения включает параметры `pki-auth-challenge` и `pki-auth-challenge-encrypted`, которые переносят значения случайно генерируемого вызова *RAND* и его зашифрованное N_{pr}*[RAND]*, соответственно. Заголовок `Content-Type` должен быть установлен в `application/x-www-form-urlencoded`.

5) Функция конечного пользователя:

- получает значение *A*, которое предположительно равно *RAND*, и значение *B*, которое предположительно равно $N_{pr}[RAND]$;
- получает сетевой открытый ключ N_{pu};
- дешифрует *B* с помощью N_{pu} и сравнивает результат с *A*. Если значения идентичны, это значит, что сеть аутентифицирована, если не идентичны, процедура аутентификации прерывается;
- генерирует секретный ключ *K*;
- генерирует значение $U_{pr}[N_{pu}[K]|K(RAND)]$, устанавливает параметр `pki-auth-user-signature` в это значение и направляет его в теле сообщения HTTP POST объекту A-2. Заголовок `Content-Type` сообщения должен быть установлен в значение `application/x-www-form-urlencoded`.

После этого шага A-2 проверяет, является ли достоверным ответ. Для этого A-2:

- запрашивает сертификат пользователя для получения открытого ключа пользователя U_{pu};
- дешифрует с помощью U_{pu} полученное значение *C*, которое предположительно равно $U_{pr}[N_{pu}[K]|K(RAND)]$, для получения значения *D*|*E*, где *D* предположительно равно $N_{pu}[K]$, а *E* предположительно равно *K(RAND)*;
- дешифрует с помощью сетевого открытого ключа N_{pr} значение *D* для получения *K′*;
- дешифрует с помощью *K′* значение *E* для получения *RAND′*;
- сравнивает *RAND′* с *RAND*. Их совпадение означает, что пользователь прошел аутентификацию и *K′* = *K*. То есть функция конечного пользователя и A-2 теперь совместно используют ключ *K*.

6) Если все вышеуказанные шаги выполнены успешно, A-2 выполняет следующие операции:

- генерирует утверждение SAM, устанавливая атрибут методы элемента <SubjectConfirmation> в значение "отправитель ручается";
- вычисляет значение $K_s(K)$;
- включает это утверждение в ответ SAML. Затем он доставляет ответ SAML и рассчитанное значение $K_s(K)$ через HTTP тем же способом, что и описанный для запроса SAML на шаге 2 (то есть как часть строки запроса). Значение $K_s(K)$ переносится параметром `pki-auth-keyinfo`;
- для обеспечения аутентификации источника и целостности зашифрованного в URL сообщения, A-2 подписывает его, как описано в пункте 10.2.4.5.2 "Аспекты безопасности" [ITU-T X.1141]. Для подписания следует использовать K_s;

  после валидации подписанного URL сервер AS уверен, что утверждение SAML сделано объектом A-2. AS проверяет само утверждение (например, чтобы убедиться, что условия выполняются). После этого AS извлекает значение $K_s(K)$ и дешифрует его с помощью общего K_s для получения *K*. В этой точке AS аутентифицирует функцию конечного пользователя и оба объекта совместно используют ключ *K*, который может использоваться для защищенной связи между ними.

7) AS, если это требуется политикой для принятия решения об авторизации, получает информацию о контексте аутентификации. В этом случае ответ A-2 содержит информацию, определенную классом контекста аутентификации в разделе *Открытый ключ* – МСЭ-T X.509 [ITU-T X.1141].

8) AS направляет функции конечного пользователя результат решения об авторизации.

6.2.7.4 Дополнительные требования к объектам, участвующим в аутентификации

Для поддержки описанного механизма участвующие объекты должны отвечать следующим требованиям:

6.2.7.4.1 Требования к функции конечного пользователя

Функция конечного пользователя должна обладать возможностью:

- запускать приложение клиента HTTP;
- безопасно сохранять свой открытый U_{pr} (например, на смарт-карте);
- получать сетевой открытый ключ N_{pu};
- выполнять шифрование и дешифрование;
- генерировать ключ K.

6.2.7.4.2 Требования к серверу приложений (AS)

- AS должен поддерживать SAML [ITU-T X.1141].
- AS должен иметь разделенный секрет (K_s) с A-2.

6.2.7.4.3 Требования к функциональному объекту A-2

Функциональный объект A-2 должен обладать возможностью:

- поддерживать протокол HTTP;
- безопасно сохранять свой личный ключ N_{pr};
- получать ключ профиля пользователя U_{pu};
- выполнять шифрование и дешифрование;
- генерировать случайный вызов *RAND*;
- поддерживать SAML [ITU-T X.1141];
- иметь разделенный секрет (K_s) с AS.

6.2.7.4.4 Требования к функциональному объекту S-5

Функциональный объект S-5 должен быть в состоянии сохранять сертификаты пользователей МСЭ-Т X.509 или получать эти сертификаты из хранилища (или и то и другое).

6.2.7.5 Дополнительные требования к интерфейсам между участвующими объектами

К интерфейсам применяются следующие требования:

- интерфейс между функцией конечного пользователя и сервером приложений должен поддерживать протокол HTTP [b-IETF RFC 2616];
- интерфейсы между функцией конечного пользователя и функциональными объектами A-2 должны поддерживать протокол HTTP [b-IETF RFC 2616];
- интерфейс между A-2 и сервером приложений должен поддерживать SAML [ITU-T X.1141];
- интерфейс между функциональными объектами A-2 и S-5 должен поддерживать механизм запрос-ответ, который позволяет A-2 получать сертификаты пользователя X.509 от S-5.

6.2.8 Объединение аутентификации на базе OpenID и аутентификации с использованием AKA

Объединение аутентификации IMS и аутентификации на базе OpenID позволяет комбинировать возможности *ориентированного на сеть* и *ориентированного на пользователя* IdM. Этот механизм объединения:

- обеспечивает сетевым поставщикам возможность предоставления услуг определения идентичности пользователям, которые обращаются к веб-приложениям;
- может использоваться для обеспечения однократной регистрации (SSO) пользователей в сети IMS и среде веб-услуг с существующим приложением ISIM, а также с другими приложениями SIM, которые опираются на AKA;
- дает пользователям возможность управления своими открытыми идентификаторами в Сети, как это определено в [b-OpenID v.2], и при этом максимально использовать услуги СПП;
- повышает уровень безопасности пользователя благодаря участию пользующегося доверием пользователя поставщика сети в управлении доступом к веб-приложениям.

В [b-3GPP TR 33.924] описаны несколько решений по объединению OpenID и AKA, которые опираются на использование функции сервера начальной загрузки (BSF).

В данном разделе описан дополнительный механизм объединения OpenID и AKA. Для этого в спецификации OpenID предусмотрены разнообразные механизмы аутентификации.

OpenID может взаимодействовать с другими технологиями, такими как OAuth, как показано в Дополнении II.

6.2.8.1 Объекты, участвующие в аутентификации, и информационный поток

- Функция конечного пользователя. Этот объект обладает возможностью запуска приложения веб-клиента и осуществления связи с надлежащим приложением SIM.
- Сервер приложений — объект, обеспечивающий веб-услугу. Он играет роль полагающейся стороны.
- A-2 – функциональный объект шлюза приложений (APL-GW-FE), который может действовать как поставщик данных идентичности OpenID [b-OpenID v.2]. (A-2 необязательно разделяет кратковременный секрет с сервером приложений согласно [b-OpenID v.2]).
- S-5 – функциональный объект профиля пользователя услуги (SUP-FE).

Информационный поток процедуры аутентификации показан на рисунке 8. Процедура установления кратковременного ключа для подписания между сервером приложений и A-2 на рисунке не показана. Цифры указывают основные шаги процедуры для двух вариантов OpenID:

a) A-2 и сервер имеют разделенный секрет;

b) A-2 и сервер не имеют разделенного секрета.

Общими шагами для обоих вариантов являются шаги 1–6. Шаг 7a относится только к варианту a).

Шаги 7b, 8b и 9b относятся только к варианту b).

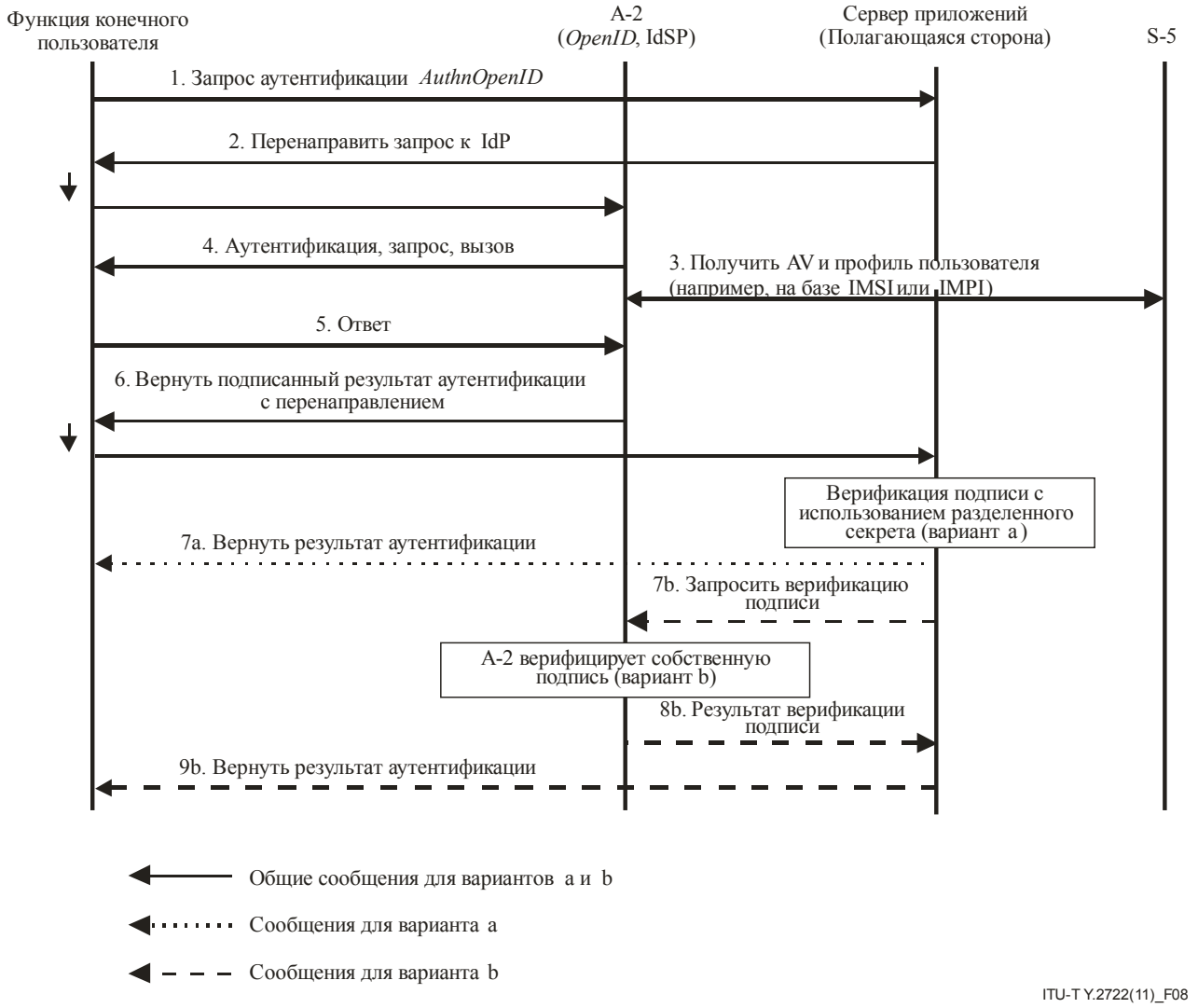


Рисунок 8 – Объединение механизмов аутентификации с использованием AKA и OpenID

Ниже описаны основные шаги:

1) Веб-клиент функции конечного пользователя выдает серверу приложений запрос аутентификации *AuthnOpenID*. Этот запрос включает идентификатор OpenID.

2) Сервер приложений, используя представленный идентификатор OpenID, обнаруживает URL объекта A-2, который действует как поставщик данных идентичности OpenID, и перенаправляет запрос аутентификации к этому URL.

   После этого шага A-2 сопоставляет идентификатор пользователя с надлежащей идентичностью (например, IMSI или IMPI).

3) A-2 получает от S-5 вектор аутентификации AKA – AV – и профиль пользователя на базе IMPI.

4) A-2 направляет функции конечного пользователя запрос аутентификации, используя протокол HTTP Digest AKA (метод дайджеста HTTP на базе AKA) [b-IETF RFC 4169] или [b-IETF RFC 3310]. Запрос включает вызов и количественное значение, которое позволяет функции конечного пользователя аутентифицировать сеть.

   После этого шага функция конечного пользователя аутентифицирует сеть, как описано в [b-IETF RFC 4169] или [b-IETF RFC 3310].

5) Функция конечного пользователя направляет ответ на запрос объекту A-2 согласно [b-IETF RFC 4169] или [b-IETF RFC 3310].

После этого шага A-2 аутентифицирует функцию конечного пользователя согласно [b-IETF RFC 4169] или [b-IETF RFC 3310].

6) A-2 направляет функции конечного пользователя подписанное сообщение, в котором утверждается, что предъявленный идентификатор OpenID принадлежит пользователю. Сообщение подписывается с использованием секрета, разделенного с сервером приложений, для варианта a). Для варианта b) сообщение подписывается с использованием секретного ключа A-2. Это сообщение включает запрос на перенаправление веб-клиента функции конечного пользователя к серверу приложений. Процедуры подписания и перенаправления подробно описаны в [b-OpenID v.2]. В [b-OpenID v.2] также определены меры для предупреждения атак, основанных на повторном использовании подписанного утверждения аутентификации.

Шаги, которые относятся к варианту a):

7a После верификации подписи ответа, полученного на шаге 6, сервер приложений извещает функцию конечного пользователя о результате аутентификации. Для этой верификации сервер приложений использует секрет, разделенный с A-2.

Если на каком-либо из шагов – с 1 по 6 или 7a – возникает ошибка, процедура аутентификации останавливается.

Шаги, которые относятся к варианту b):

7b Сервер приложений направляет копию сообщения, полученного на шаге 6, объекту A-2 с запросом верификации подписи.

8b После верификации своей собственной подписи A-2 сообщает результат верификации серверу приложений.

9b Сервер приложений сообщает результат аутентификации функции конечного пользователя.

Если на каком-либо из шагов – с 1 по 6, 7b, 8b или 9b – возникает ошибка, процедура аутентификации останавливается.

6.2.8.2 Дополнительные требования к объектам, участвующим в аутентификации

Для поддержки описанного механизма участвующие объекты должны отвечать следующим требованиям:

6.2.8.2.1 Требования к функции конечного пользователя

Функция конечного пользователя должна обладать возможностью:

- выполнять аутентификацию с использованием протокола HTTP Digest AKA;
- осуществлять связь с надлежащим приложением SIM.

6.2.8.2.2 Требования к серверу приложений

Сервер приложений должен иметь возможность поддерживать спецификацию OpenID версии 2.0 [b-OpenID v.2].

6.2.8.2.3 Требования к функциональному объекту A-2

Функциональный объект A-2 должен обладать возможностью:

- выполнять аутентификацию дайджеста HTTP на базе AKA (HTTP Digest AKA);
- сопоставлять идентификатор пользователя OpenID с надлежащим идентификатором (таким как IMSI или IMPI);
- действовать как поставщик данных идентичности OpenID.

6.2.8.2.4 Требования к функциональному объекту S-5

К функциональному объекту S-5 не предъявляется дополнительных требований, кроме описанных в [ITU-T Y.2012].

6.2.8.3 Дополнительные требования к интерфейсам между участвующими объектами

К интерфейсам применяются следующие требования:

- интерфейс между функцией конечного пользователя и сервером приложений должен поддерживать аутентификацию OpenID, как определено в версии 2.0 спецификации [b-OpenID v.2];
- интерфейс между функцией конечного пользователя и функциональными объектами A-2 должны поддерживать протокол HTTP Digest AKA [b-IETF RFC 4169] или [b-IETF RFC 3310];

Интерфейс между функциональными объектами A-2 и S-5 не должен обусловливать каких бы то ни было требований, присущих механизму.

6.2.8.4 Механизм взаимодействия OpenID и AKA для сценария разделения терминала пользователя

Механизм, описанный в данном пункте, поддерживает также сценарий разделения терминала, который изложен в [b-3GPP TR 33.924]. Сценарий разделения терминала пользователя связан с ситуацией, в которой осуществляющий аутентификацию агент (объект, имеющий доступ к карте UICC) и осуществляющий просмотр агент находятся в разных терминалах пользователя.

Учитывая, что в прямом решении AKA, описанном в данном пункте, IdSP соответствует свернутым функциям NAF/BSF, сценарии, описанные в [b-3GPP TR 33.924], полностью поддерживаются этим решением. В основе механизма лежит прямая аутентификация с использованием AKA, а не аутентификация на основе GBA.

6.2.9 Общая архитектура начальной загрузки (GBA)

Общая архитектура начальной загрузки (GBA) определяет рамки аутентификации начальной загрузки и установления соглашения о ключе, используя механизм соглашения об аутентификации и ключе (AKA) 3GPP. GBA упрощает аутентификацию конечных пользователей для сетевой прикладной функции (NAF) и может использоваться в управлении определением идентичности в СПП для обеспечения:

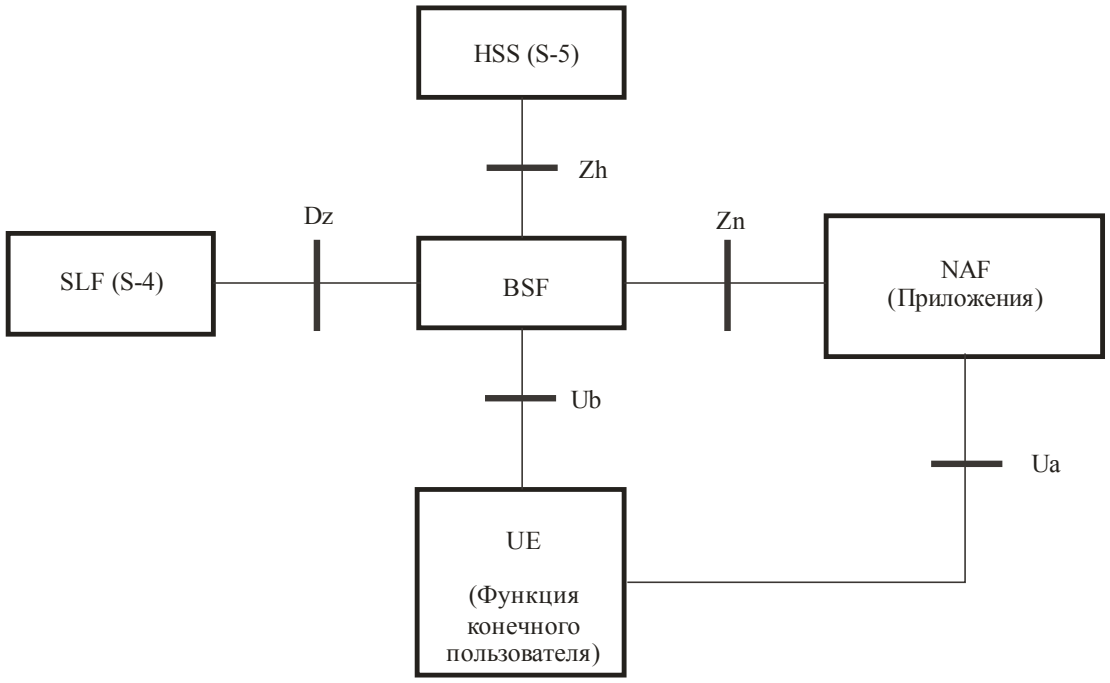
- соглашения об аутентификации и ключе;
- неприкосновенности частной жизни;
- однократной регистрации.

GBA – это система аутентификации, включающая следующие три стороны:

- конечный пользователь, который пытается получить доступ к услугам сети, используя для этого оборудование пользователя (UE);
- сервер приложений (называемый сетевой прикладной функцией или NAF);
- доверенный объект (называемый функцией начальной загрузки сервера или BSF), который участвует в аутентификационном обмене и обмене ключами между двумя другими объектами.

GBA делает возможной аутентификацию конечного пользователя, использующего UE, по отношению к серверу приложений (NAF), не раскрывая NAF долговременные полномочия и секреты конечного пользователя, используя BSF доверенного объекта.

На рисунке 9 изображена GBA [b-ETSI TS 133 220] и представлено преобразование объектов, определенных 3GPP, в функциональные объекты, определенные в [ITU-T Y.2012].



ПРИМЕЧАНИЕ. – Метки в скобках означают объекты, определенные в [ITU-T Y.2012].

ITU-T Y.2722(11)_F09

Рисунок 9 – Простая сетевая модель для начальной загрузки

Процедура GBA включает следующие основные шаги:

1) NAF запрашивает аутентификацию и согласовывает использование GBA через опорную точку Ua.

2) Клиент функции BSF, работающий на UE, инициирует процедуру начальной загрузки через опорную точку Ub. BSF забирает аутентификационную информацию и установки безопасности пользователя GBA из системы HSS через Zh. UE и BSF выполняют взаимную аутентификацию, используя протокол HTTP Digest AKA. Результатом процедуры является получение UE идентификатора транзакции начальной загрузки (B-TID) от функции BSF и установление совместного ключа (Ks), который используют оборудование UE и функция BSF.

3) UE выводит Ks_NAF из Ks и направляет B-TID (вместе с обусловливаемыми приложением данными) функции NAF.

4) NAF направляет B-TID функции BSF через опорную точку Zn.

5) На основании B-TID функция BSF определяет подлежащий использованию Ks, выводит из него Ks_NAF и направляет Ks_NAF функции NAF.

6) Наконец, UE и NAF могут выполнить взаимную аутентификацию, используя совместный ключ Ks_NAF. Точная процедура аутентификации зависит от протокола, используемого между UE и NAF. Например, GBA определяет, что приложения на базе HTTP могут использовать либо аутентификацию по протоколу HTTP Digest [b-IETF RFC 2617], либо шифронаборы заранее установленных совместных ключей TLS [b-IETF RFC 4279].

ПРИМЕЧАНИЕ. – Функция BSF обращается с запросом к SLF через опорную точку Dz для получения имени HSS, содержащего определяемые абонентами данные. Функция SLF не требуется, если конфигурация BSF предусматривает использование предопределенной HSS.

Отображение объектов GBA в объекты СПП, которое определено в [ITU-T Y.2012] "Функциональные требования и архитектура сетей последующих поколений", является следующим:

- NAF соответствует объекту "Приложения" на рисунке 3 – "Обобщенная функциональная архитектура СПП" ([ITU-T Y.2012]).
- HSS соответствует функциональному объекту профиля пользователя услуги S-5.
- SLF соответствует функциональному объекту указателя контракта S-4.
- UE соответствует функции конечного пользователя.

6.2.10 Аутентификация на базе IMSI

В зависимости от требуемого уровня гарантии для аутентификации, обеспечивающей обратную совместимость, в сети на основе протокола беспроводных приложений (WAP) может использоваться международный идентификатор абонента подвижной связи (IMSI). Поскольку IMSI является уникальной текстовой строкой, она может использоваться в качестве идентичности объекта для конкретной услуги.

Этот подход:

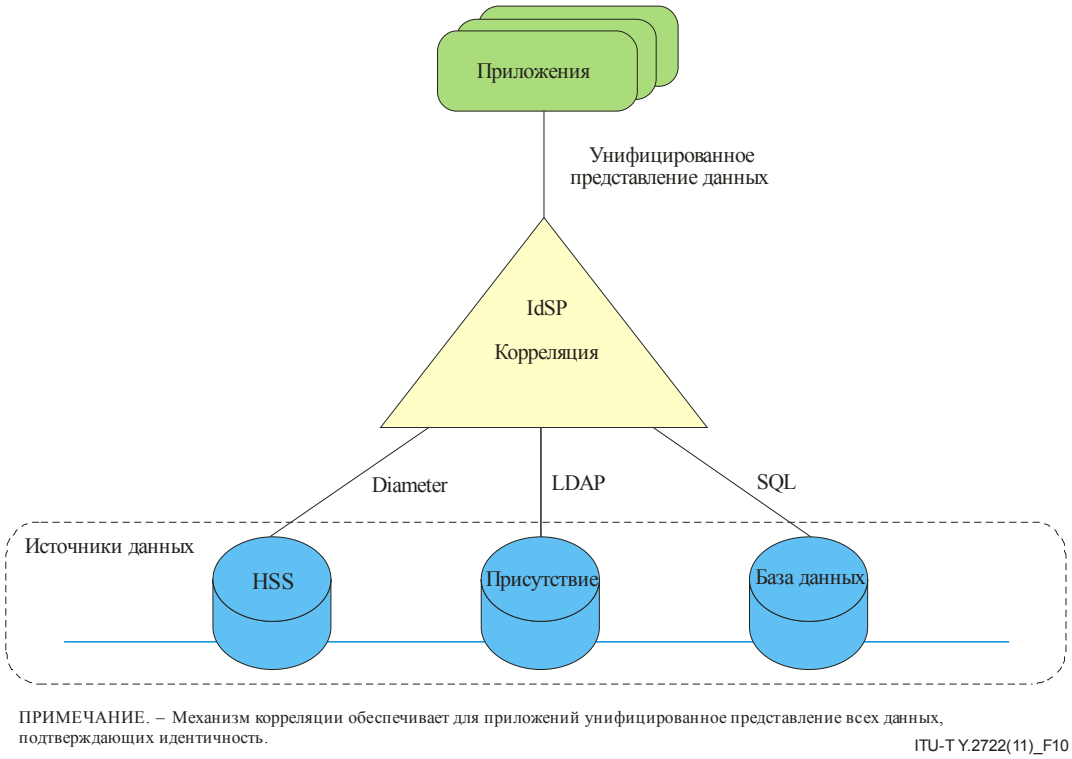
- использует код IMSI как идентичность объекта в беспроводных приложениях;
- обеспечивает надежный канал услуги при условии, что конечная точка имеет законный IMSI, когда конечная точка делает запрос на аутентификацию;
- предусматривает, что все системы доверяют результату аутентификации шлюза WAP и обеспечивают услуги для этого объекта;
- может использоваться для обеспечения функции однократной регистрации посредством уникальности IMSI для той же конечной точки между GPRS/CDMA 1x и беспроводными приложениями (например, беспроводным почтовым ящиком и т. д.);
- обеспечивает защиту кода IMSI.

6.3 Корреляция и увязка

В [ITU-T Y.2720] "Структура управления определением идентичности в СПП" указано, что *информация, подтверждающая идентичность (например, идентификаторы, регистрационные данные и атрибуты), может сопоставляться для создания увязки с целью гарантирования идентичности объекта*.

Цель решения, которое делает возможной корреляцию, заключается в сборе разных типов информации, подтверждающей идентичность, из разных источников и представление этой информации приложениям в понятном для них унифицированном формате.

Концепция такого решения показана на рисунке 10. На рисунке показаны три примера источников информации, подтверждающей идентичность: HSS, сервер присутствия и база данных определяемых приложением пользовательских данных. Для принятия решений об аутентификации и авторизации приложению могут потребоваться три типа информации. В представленном примере для получения данных из соответствующих источников механизм корреляции использует протоколы Diameter, LDAP и SQL. Эти данные далее представляются приложению в понятном для него формате. Таким образом, механизм корреляции освобождает приложение от нагрузки, связанной с поддержкой нескольких протоколов для связи с разными источниками информации, подтверждающей идентичность.



ПРИМЕЧАНИЕ. – Механизм корреляции обеспечивает для приложений унифицированное представление всех данных, подтверждающих идентичность.

ITU-T Y.2722(11)_F10

Рисунок 10 – Корреляция информации, подтверждающей идентичность

6.4 Обнаружение

В [ITU-T Y.2721] "Требования к управлению определением идентичности в СПП и сценарии использования" определяется, что СПП/IdSP должны поддерживать функции и возможности обнаружения источников информации, подтверждающей идентичность, в пределах домена СПП/IdSP между разными доменами СПП/IdSP.

В данном пункте представлены примеры стандартных механизмов, которые поддерживают эти требования, и даются ссылки на соответствующие спецификации.

6.4.1 Внутрисетевое обнаружение

В [ITU-T Y.2012] "Функциональные требования и архитектура СПП варианта 1" определяется специальный объект – функциональный объект указателя контракта (SL-FE), – который предоставляет адрес функционального объекта профиля пользователя услуги (SUP-FE), хранящего подтверждающую идентичность информацию конкретного абонента. SL-FE делает возможным обнаружение SUP-FE, который отвечает за хранение профилей пользователя, данных о местоположении, относящихся к абоненту, и данных о статусе присутствия. Направляя запросы к SUP-FE, объекты сети могут получить эту информацию, подтверждающую идентичность. Как указано в [ITU-T Y.2012], направлять запрос к SL-FE для получения адреса соответствующего SUP-FE могут следующие сетевые объекты:

- функциональный объект поддержки приложений (AS-FE);
- функциональный объект запрашивающей функции управления сеансами связи (I-CSC-FE);
- функциональный объект обслуживающей функции управления сеансами связи (S-CSC-FE).

Механизм, позволяющий этим объектам осуществлять поиск в сети оператора адреса SUP-FE, который хранит подтверждающую идентичность информацию для данного пользователя, определен в [3GPP TS 23.228]. Следует заметить, что объекты [ITU-T Y.2012] отображаются в объекты [3GPP TS 23.228] следующим образом:

- AS-FE соответствует AS;
- I-CSC-FE соответствует I-CSCF;
- S-CSC-FE соответствует S-CSCF;
- SL-FE соответствует SLF.

6.4.2 Межсетевое обнаружение

Примеры механизмов межсетевого обнаружения IdSP включают примеры, определенные в SAML [ITU-T X.1141] и ID-WSF [b-LA WSF]. Эти механизмы зависят от предварительно достигнутого соглашения между участвующими объектами (например, IdSP и полагающейся стороной) или членами федерации.

Еще одним примером является OpenID [b-OpenID v.2], устанавливающий механизм обнаружения, который позволяет полагающейся стороне определять местоположение IdSP пользователя на основе предоставляемого пользователем идентификатора OpenID.

6.5 Передача информации IdM и обмен этой информацией

В данном пункте рекомендуются протоколы и механизмы для передачи информации, подтверждающей идентичность, и обмена этой информацией.

6.5.1 Безопасность передачи и обмена IdM

В данном пункте рекомендуются протоколы обеспечения защиты целостности и конфиденциальности при передаче IdM.

6.5.1.1 Решения на базе SAML 2.0 [ITU-T X.1141]

Для защиты целостности и конфиденциальности SAML 2.0 рекомендует использовать протокол, обеспечивающий безопасную передачу по каналу или сети, такой как TLS или IPSec, в конфигурации которого предусмотрена защита пакетов, передаваемых через это сетевое соединение.

Для защиты целостности на уровне сообщений в дополнение к защищенному каналу связи может использоваться подпись XML. При использовании подписи XML требуется следовать положениям пункта 8.4 "Синтаксис и обработка подписи SAML и XML" [ITU-T X.1141].

Для защиты конфиденциальности на уровне сообщений в дополнение к защищенному каналу связи может использоваться шифрование XML. При использовании шифрования XML требуется следовать положениям пункта 8.4 "Синтаксис и обработка подписи SAML и XML" [ITU-T X.1141].

6.5.1.2 Структура веб-услуг на основе идентичности (известная как ID-WSF)

Для использования ID-WSF предполагается, что целостность ее передач и сообщений, которыми обмениваются отправитель и получатель, защищены. Аналогично SAML 2.0 этим стандартом рекомендуется использовать протокол, обеспечивающий безопасную передачу по каналу или сети, такой как TLS или IPSec, в конфигурации которого предусмотрена защита пакетов, передаваемых через это сетевое соединение [b-LA ID-WSF security].

1) Защита канала на транспортном уровне

В случае использования SSL или TLS в качестве защищенного сетевого протокола для ID-WSF рекомендуется использовать протоколы SSL 3.0, TLS 1.0 либо более высокой версии. Объект, завершающий соединение SSL (3.0) или TLS (1.0) должен предлагать или принимать в процессе рукопожатия подходящие шифронаборы. Ниже приведен неисчерпывающий перечень рекомендуемых шифронаборов TLS 1.0 (или его эквивалента SSL 3.0).

- TLS_RSA_WITH_RC4_128_SHA;
- TLS_RSA_WITH_3DES_EDE_CBC_SHA;
- TLS_DHE_DSS_WITH_3DES_EDE_CBC_SHA;
- TLS_RSA_WITH_AES_CBC_SHA;
- TLS_DHE_DSS_WITH_AES_CBC_SHA.

Для подписания и верификации протокольных сообщений рекомендуется, чтобы осуществляющие связь объекты использовали сертификаты и личные ключи, отличающиеся от сертификатов и личных ключей, которые применялись для защиты канала SSL или TLS.

Другие протоколы безопасности, такие как IPSec или Kerberos, могут использоваться при условии, что они реализуют эквивалентные меры защиты.

2) Защита конфиденциальности сообщений

При наличии посредников осуществляющие связь объекты должны обеспечивать, чтобы конфиденциальная информация не была раскрыта не имеющим к ней доступа объектам. В этом случае требуется, чтобы эти объекты использовали механизмы конфиденциальности, определенные в спецификации "Безопасность веб-услуг (WSS): Безопасность сообщений SOAP" OASIS [b-OASIS WSS SOAP], для шифрования содержимого конверта SOAP <S:Body>.

3) Правила целостности сообщений

Правила целостности сообщений в данном пункте применяются, только если используется спецификация "Безопасность веб-услуг (WSS): Безопасность сообщений SOAP" OASIS [b-OASIS WSS SOAP] для сообщения по протоколу ID-WSF, связанного с SOAP согласно Liberty SOAP Binding версии 2.0 [b-LA SOAP binding].

В этом случае требуется, чтобы отправитель создал единичную подпись <ds:Signature>, содержащуюся в заголовке <wsse:Security>, и эта подпись должна иметь ссылку на все компоненты сообщения, которые должны быть подписаны.

В частности, эта подпись должна иметь ссылку на элемент тела SOAP (сам элемент), жетон безопасности, связанный с этой подписью, и все заголовки в этом сообщении, которые были определены в Liberty SOAP Binding версии 2.0 [b-LA SOAP binding], включая обязательный и необязательный блоки заголовка.

Примером жетона безопасности является элемент <saml2:Assertion>, передаваемый в заголовке <wsse:Security>. Заголовок wsu:Timestamp в блоке заголовка wsse:Security, блоки заголовка wsa:MessageID, wsa:RelatesTo, sb:Framework, sb:Sender и sb:InvocationIdentity служат примерами элементов заголовка, на которые будет сделана ссылка в подписи.

Следует заметить, что необходимо соблюдать осторожность при формировании элементов, содержащихся в параметрах ссылки на конечную точку, поскольку они будет переведены в блоки заголовка SOAP. Следует принимать меры во избежание конфликта или дублирования атрибутов идентификаторов, например путем применения методов генерирования уникальных идентификаторов.

Если сообщение подписано, отправитель должен включить результирующую подпись XML в элемент <ds:Signature> как порождение заголовка <wsse:Security>.

Элемент <ds:Signature> должен содержать ссылку на ключ подтверждения субъекта с элементом <ds:KeyInfo>. Элемент <ds:KeyInfo> должен быть включен в элемент <wsse:SecurityTokenReference>, так чтобы ключ подтверждения субъекта мог быть найден в пределах заголовка <wsse:Security>. Включение ссылки рекомендуется для выполнения руководящих указаний, содержащихся в пункте 3.4.2 спецификации "Безопасность веб-услуг: Профиль жетона SAML 1.1" OASIS [b-OASIS SAML token].

i) Правила обработки для отправителя:

- Построение и оформление элемента заголовка <wsse:Security> требуется для выполнения правил, описанных в спецификации "Безопасность веб-услуг: Профиль жетона SAML 1.1" OASIS [b-OASIS SAML token].
- Элемент заголовка <wsse:Security> должен иметь атрибут mustUnderstand, логическое значение которого истина.
- Отправитель должен поместить жетон безопасности аутентификации сообщения как прямое порождение элемента <wsse:Security>.
- Если используются механизмы аутентификации сообщения, отправитель должен выполнять правила целостности сообщений, составленные для отправителей и получателей.

К жетонам канала следующие соображения не применяются:

- Для использования установок, которые требуют независимой аутентификации сообщений, требуется, чтобы обязательство было выполнено путем подписания тела сообщения и частей заголовка и помещения <ds:Signature> как прямого порождения заголовка <wsse:Security>.
- Для использования установок, которые не требуют независимой аутентификации сообщений, обязательство по подтверждению субъекта может быть выполнено путем сопоставления сертификата и ключа, используемых для воздействия на аутентификацию равноправного объекта с сертификатом и ключом, которые описываются в жетоне аутентификации сообщения. Для обеспечения этого орган, выпускающий утверждения, должен сформировать такое утверждение, чтобы ключ подтверждения мог быть однозначно верифицирован как тот же сертификат и ключ, которые использовались при выполнении аутентификации равноправного объекта. Это необходимо для снижения риска угрозы атаки на основе подмены сертификата. Рекомендуется, чтобы сертификат или цепочка сертификатов были связаны с ключом подтверждения субъекта.

ii) Правила обработки для получателя

- Получатель должен определять местоположение элемента <wsse:Security>, адресатом которого он является. При этом ДОЛЖНЫ соблюдаться правила, описанные в спецификация "Безопасность веб-услуг (WSS): Безопасность сообщений SOAP" OASIS [b-OASIS WSS SOAP], и применимые профили жетонов WSS (например, "Безопасность веб-услуг: Профиль жетона SAML 1.1" OASIS [b-OASIS SAML token] для жетонов SAML).
- Элемент заголовка <wsse:Security> должен иметь атрибут mustUnderstand, логическим значением которого является истина, а получатель должен иметь возможность обработать этот блок заголовка согласно спецификации "Безопасность веб-услуг (WSS): Безопасность сообщений SOAP" OASIS [b-OASIS WSS SOAP] и соответствующим профилям жетонов WSS (например, "Безопасность веб-услуг: Профиль жетона SAML 1.1" OASIS [b-OASIS SAML token] для жетонов SAML).
- Получатель должен определить местоположение жетона безопасности и получатель должен определить, что он доверяет органу, выпустившему этот жетон.
- Получатель должен проверить достоверность подписи выпустившего жетон органа на жетоне. Эта проверка достоверности необходима для выполнения основных правил проверки, описанных во втором издании документа по *синтаксису и обработке подписи XML* консорциума Всемирной паутины (W3C) [b-W3C XML signature]. Рекомендуется, чтобы получатель проверил семантику доверия ключа подписания ввиду риска некорректной аутентификации.
- Если сообщение было подписано, получатель должен определить местоположение элемента <ds:Signature>, переносимого внутри заголовка <wsse:Security>.
- Если механизмом безопасности является не peerSAMLV2, получатель должен принять решение относительно содержимого элемента <ds:KeyInfo>, переносимого в <ds:Signature>, и использовать описывающий его ключ для проверки достоверности подписанных элементов. Если механизмом безопасности является peerSAMLV2, ключ является ключом клиента, используемым в аутентификации клиента SSL/TLS.
- Если используются механизмы аутентификации, получатель должен следовать правилам целостности сообщения, составленным для отправителей и получателей.

4) Обработка сообщений с жетоном WSS МСЭ-Т X.509

В данном пункте описаны правила семантики и обработки для механизмов сообщений, имеющих значение МСЭ-Т X.509. Пример приведен в Дополнении I.

Идентификаторы URI поддерживают одностороннюю (отправитель) аутентификацию сообщений и имеют следующую форму:

- *urn:liberty:security:2003-08:PEER:X509*, где PEER может изменяться в зависимости от используемого механизма аутентификации однорангового объекта (например, может быть нулевым, TLS и т. д.).

Механизм аутентификации сообщений WSS МСЭ-Т X.509 использует профиль жетона сертификации МСЭ-Т X.509 безопасности сетевых услуг [b-OASIS WSS X.509 profile] как средство, с помощью которого отправитель сообщения аутентифицирует получателя. Эти механизмы аутентификации сообщений являются односторонними. Это означает, что аутентифицируется только отправитель сообщения. В сферу действия настоящей Рекомендации не входит предложение о том, когда должны аутентифицироваться ответные сообщения, однако стоит заметить, что этот механизм можно также положить в основу аутентификации ответных сообщений. Рекомендуется признать, вместе с тем, что независимая аутентификация ответных сообщений не обеспечивает такую же семантику защиты потока сообщений, которую обеспечил бы механизм взаимной аутентификации одноранговых объектов.

Для использования установок, которые требуют аутентификацию сообщений независимо от аутентификации одноранговых объектов, требуется, чтобы отправляющий сообщение одноранговый объект выполнял аутентификацию сообщений, предъявляя доказательство владения ключом, связанным с жетоном МСЭ-Т X.509. Этот ключ должен признаваться получателем, как принадлежащий отправляющему сообщение одноранговому объекту.

Если отправитель владеет ключом подтверждения субъекта для подписи элементов сообщения, эта подпись гарантирует подлинность и целостность охваченных ею элементов. Вместе с тем одна эта мера не смягчает угрозы повторной передачи, внесения ложной информации и определенных классов атак с целью изменения сообщений. Для защиты сообщений от таких угроз, может использоваться один из механизмов, поддерживающих аутентификацию одноранговых объектов, или необходима базовая модель обработки запросов связи SOAP.

i) Правила обработки для отправителя

Изложенные в данном пункте правила являются дополнительными к общим правилам обработки аутентификации сообщений, описанным в настоящей Рекомендации.

- Отправитель должен продемонстрировать владение личным ключом, связанным с подписью, сформированной в сочетании с профилем жетона WSS МСЭ-Т X.509.
- Для применения установок, которые требуют независимой аутентификации сообщений, это обязательство должно быть выполнено путем подписания частей сообщения, при необходимости, и записи информации в заголовок <wsse:Security> (согласно [b-OASIS WSS SOAP]).
- Для использования установок, которые не требуют независимой аутентификации сообщений, отправитель должен выполнить это обязательство путем добавления к заголовку безопасности элемента <ds:KeyInfo>, который переносит сертификат.

Требуется, чтобы этот сертификат был однозначно верифицирован как тот же сертификат и ключ, которые использовались при осуществлении аутентификации однорангового объекта. Это необходимо для снижения риска угрозы атаки с целью подмены сертификата. Более того, следует отметить, что эта оптимизация применяется только к механизмам *ClientTLS:X509*.

ii) Правила обработки для получателя

- Если политика проверки подлинности, которая касается аутентификации одноранговых объектов, считается достаточной для целей аутентификации, то получатель должен установить соответствие сертификата и ключа, которые использовались для осуществления аутентификации одноранговых объектов, с соответствующей информацией о ключе, переносимой в этом сообщении. Это позволяет получателю сообщения определять, что отправитель сообщения предполагал использование конкретной идентичности аутентифицированного транспорта. Информация, связанная с ключом SSL/TLS для сообщения МОЖЕТ переноситься в сообщении, используя жетон безопасности МСЭ-Т X.509 безопасности сообщений OASIS SOAP.

6.6 Защита информации, позволяющей установить личность (PII)

Согласно [ITU-T Y.2720] "Структура управления определением идентичности в СПП" защита PII является вопросом национального и регионального регулирования. Хотя механизмы и процедуры, используемые для поддержки защиты PII, могут меняться в зависимости от системы регулирования, в их основе лежат общие принципы.

В данном пункте содержится краткий обзор процедур защиты PII, определенных в специальном докладе NIST "Руководство по защите конфиденциальности информации, позволяющей установить личность" [b-NIST-SP 800-122]. Содержащиеся в этом документе спецификации *мер защиты конфиденциальности PII*, могут использоваться в качестве руководящих указаний разработчиками систем IdM. Определяются следующие категории мер защиты:

- Эксплуатационные меры защиты
  - разработка политики и процедур;
  - осведомленность, подготовка и обучение.
- Меры защиты, обусловливаемые неприкосновенностью частной жизни
  - предельное уменьшение использования, сбора и хранения PII;
  - проведение оценок воздействия на конфиденциальность;
  - обезличивание информации;
  - анонимизация информации.
- Средства обеспечения безопасности

  В пункте "Средства обеспечения безопасности" содержатся руководящие указания относительно механизмов и процедур обеспечения безопасности, которые не обусловливаются PII, но могут использоваться для защиты PII. Аналогично, механизмы обеспечения безопасности, определенные в [ITU-T Y.2704] и не обусловливаемые PII, могут использоваться для защиты PII.

6.7 Функции федеративной идентичности

В [ITU-T Y.2721] "Требования к управлению определением идентичности в СПП и сценарии использования" поясняется, что *общая концепция федерации состоит в том, чтобы предоставить возможность каждому члену федерации оставаться независимым и при этом содействовать обмену конкретной информацией, подтверждающей идентичность, для обеспечения возможности предоставления федеративных услуг.*

В данном пункте рекомендуется использование двух широко реализуемых стандартных механизмов, которые дают пользователю возможность получить доступ ко многим услугам без подписки на каждую услугу отдельно.

В Рекомендации SAML [ITU-T X.1141] содержится стандартное решение для федерации. Оно используется в основном коммерческими предприятиями, правительственными организациями и их поставщиками услуг.

В OpenID [b-OpenID v.2] описывается ориентированное на пользователя решение, которое широко применяется для получения доступа к веб-услугам в интернете.

6.7.1 Сопряжение и взаимодействие сетей

В настоящей Рекомендации описан ряд механизмов, которые поддерживают сопряжение и взаимодействие сетей для разных решений и федераций IdM. Основные механизмы описаны в следующих пунктах:

- объединение аутентификации на базе PKI и IMS (пункт 6.2.6);
- объединение аутентификации на базе PKI и механизмов утверждений SAML (пункт 6.2.7);
- объединение аутентификации на базе OpenID и аутентификации с использованием AKA (пункт 6.2.8);
- общая архитектура начальной загрузки (пункт 6.2.9);
- корреляция и увязка (пункт 6.3);
- функции федеративной идентичности (пункт 6.7).

6.7.2 Обнаружение IdSP в федеративной среде

В пункте 11.4.3 Рекомендации SAML [ITU-T X.1141] определяется *профиль обнаружения поставщика*, который обеспечивает для поставщика услуг возможность обнаружить поставщиков данных идентичности пользователя. Профиль определяется в поддержку *профиля SSO для веб-браузера* SAML (который определен в пункте 11.4.1 [ITU-T X.1141]).

В OpenID [b-OpenID v.2] определен механизм обнаружения, который позволяет полагающейся стороне обнаружить местоположение IdSP пользователя на основе предоставленного пользователем идентификатора OpenID identifier.

6.8 Управление доступом к информации, подтверждающей идентичность

В [ITU-T Y.2721] требуется, чтобы подтверждающая идентичность информация была доступна только авторизованным объектам, в соответствии с применимыми регулированием и политикой. В данном пункте описываются механизмы, которые могут использоваться для верификации авторизационных привилегий.

6.8.1 Базирующийся на SAML механизм совместного использования атрибутов

Утверждения SAML, в которых содержатся заявления об атрибутах, могут использоваться в качестве механизма управления привилегиями. Для распределения жетонов SAML может использоваться механизм, описанный в пункте 6.2.1.

6.8.2 Инфраструктура управления привилегиями на основе МСЭ-Т X.509

Структура сертификата атрибута, определенная в [ITU-T X.509], может использоваться в качестве механизма для инфраструктуры управления привилегиями.

6.9 Однократная регистрация

Однократная регистрация (SSO) – это функциональная возможность сети, которая позволяет пользователю однократно зарегистрироваться и получать доступ к нескольким прикладным услугам сети без повторного запроса его/ее регистрационных данных аутентификации для каждой отдельной прикладной услуги. Эта возможность существенно упрощает работу пользователя, разрешая ему получать различные услуги без многократного представления своих регистрационных данных аутентификации (например, пары имя пользователя/пароль). Поскольку SSO разрешает пользователю иметь один набор регистрационных данных аутентификации для доступа к нескольким прикладным услугам, такая регистрация облегчает поставщикам услуг возможность введения более жестких правил создания регистрационных данных. Это помогает повысить безопасность сети.

С другой стороны, взлом регистрационных данных пользователя затронет поддерживающую SSO сеть значительно сильнее, чем систему, которая не поддерживает SSO. В связи с этим важно использовать для SSO механизмы обеспечения безопасности. В данном пункте представлен обзор нескольких механизмов, которые можно использовать для поддержки SSO.

6.9.1 Механизм на базе GBA

В пункте 6.2.9 описано применение GBA для аутентификации пользователя по отношению к любой сетевой прикладной функции (NAF). Таким образом, GBA эффективно обеспечивает единый механизм регистрации пользователя для всех поддерживающий GBA функций NAF сети. Так, если пользователь зарегистрирован для работы с NAF, BSF и UE уже аутентифицировали друг друга и установили разделенный (Ks). Далее процедура регистрации пользователя для следующей NAF будет состоять из шагов 1, 3, 4, 5 и 6 (шаг 2 пропускается), описанных в пункте 6.2.9. И опять результатом этой процедуры будет секрет (Ks_NAF), разделенный между UE и новой NAF. Этот разделенный секрет может использоваться для аутентификации между UE и NAF.

SSO, базирующаяся на GBA, рекомендуется для применения в среде, в которой реализована GBA.

6.9.2 Механизм на базе SAML

Механизмы базирующейся на SAML однократной регистрации (SSO) описаны в пункте 11.4 "Профили SSO языка SAML" [ITU-T X.1141]. В этой Рекомендации определяется набор профилей SAML, поддерживающих SSO, и она содержит также *профиль единого выхода из системы* (пункт 11.4.4 [ITU-T X.1141]). В этом профиле описана процедура, которая разрешает пользователю выходить из всех приложений, для которых он/она зарегистрировался посредством SSO.

SAML v.2 предполагает доверительные отношения между IdSP и полагающимися сторонами (RP), которые устанавливаются заранее. Он также поддерживает псевдонимы-идентификаторы между IdSP и RP. Он подходит для приложений, в которых используются договорные соглашения, такие как SLA, или особо важные информация и транзакции.

6.9.3 Механизм на базе OpenID

Аутентификация OpenID версии 2.0 поддерживает возможность однократной регистрации, для того чтобы разрешить тому или иному конечному пользователю доступ к нескольким полагающимся сторонам после успешной аутентификации этого пользователя. При этом не требуется доверительных отношений между IdSP и RP. Поскольку этот механизм для идентификации пользователей поддерживает только формат на базе URL/URI, его использование требует DNS. Таким образом, он подходит для приложений на основе веб-услуг, в которых участвуют менее важные информация и транзакции.

6.10 Единый выход из системы

Протокол единого выхода из системы SAML, пункт 8.2.7 [ITU-T X.1141], разрешает конечному пользователю выходить из нескольких участвующих сеансов практически одновременно. Участвующие сеансы – это сеансы, которые были установлены через IdSP (то есть IdSP провел оценку идентичности пользователя для этих сеансов между пользователем и приложениями). IdSP отслеживает все аутентифицированные сеансы с разными полагающимися сторонами, которые пользователь установил через IdSP. Это включает аннулирование регистрационных данных аутентификации (например, cookie-файлы, утверждения) для завершенных сеансов. Этот протокол может использоваться в следующих случаях:

1) Пользователь выходит из одного из сеансов и указывает, что он/она желает выйти из всех сеансов, которые были инициированы IdSP.
2) Пользователь указывает непосредственно поставщику IdSP, что он/она желает выйти из всех сеансов.
3) IdSP выводит пользователя без его/ее запроса (например, вследствие тайм-аута).

Этот протокол определяет участвующие объекты, их поведение, потоки сообщений и формат обмениваемых сообщений. В последующих подпунктах описано использование данного протокола для перечисленных выше случаев.

6.10.1 Пользователь выходит из одного из сеансов и указывает, что он/она желает выйти из всех сеансов, которые были инициированы IdSP

На рисунке 11 показаны основные шаги потока сообщений, которые описаны ниже.

6.10.1.1 Объекты, участвующие в процедуре, и информационный поток

Участвующими являются следующие объекты:

- функция конечного пользователя;
- сервер приложений 1 (AS1) — объект, предоставляющий услугу. Он играет роль полагающейся стороны. Действует как запрашивающий и отвечающий элемент SAML, описание которого содержится в [ITU-T X.1141];
- сервер приложений 2 (AS2) — объект, предоставляющий услугу. Он играет роль полагающейся стороны. Действует как запрашивающий и отвечающий элемент SAML, описание которого содержится в [ITU-T X.1141];
- A-2 – функциональный объект шлюза приложений (APL-GW-FE), который выполняет функции IdSP и действует как запрашивающий и отвечающий элемент SAML, описание которого содержится в [ITU-T X.1141].

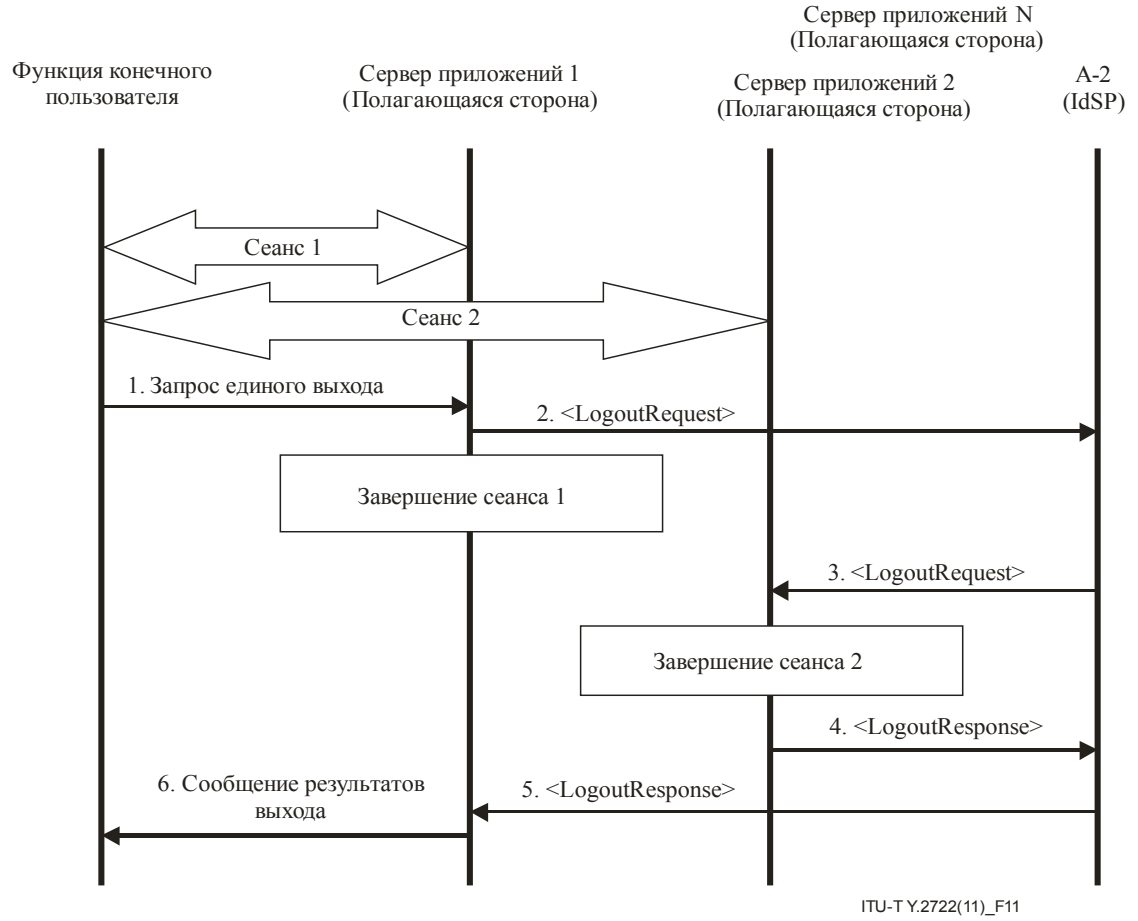


Рисунок 11 – Единый выход на базе SAML, запрашиваемый пользователем в участвующем сеансе

Основные шаги процедуры единого выхода (также называемой разрегистрацией):

1) Функция конечного пользователя вызывает запрос выхода на сервере приложений 1 (AS1), указывая, что она желает выйти из всех участвующих сеансов.

2) AS1 запрашивает выход у всех участвующих сеансов направляя запрос выхода <LogoutRequest> на A-2. Запрос должен быть подписан для аутентификации и защиты целостности согласно описанию в пункте 8.2.7 [ITU-T X.1141].

   После этого шага AS1 делает попытку завершить сеанс 1. Для этого AS1 аннулирует регистрационные данные аутентификации данного сеанса (например, утверждения, cookie-файлы), что заставит функцию конечного пользователя пройти процедуру аутентификации, если она направит другой запрос на AS1.

3) После валидации запроса от AS1, A-2 направляет сообщения <LogoutRequest> всем полагающимся сторонам (на рисунке 11 показан только AS2). Эти запросы должны быть подписаны, как определено в пункте 8.2.7 [ITU-T X.1141].

После валидации запроса выхода AS2 делает попытку завершить сеанс 2.

4) AS2 сообщает отправителю запроса на выход (A-2) результат попытки выхода, направляя сообщение <LogoutResponse>, которое должно быть подписано.

5) A-2 направляет сообщение <LogoutResponse> отправителю, инициировавшему запрос выхода (AS1), сообщая результаты единого выхода (например, успешный, частичный выход). Ответ должен быть подписан.

После этого шага A-2 обновляет свой список активных сеансов и аннулирует регистрационные данные аутентификации (например, cookie-файлы, утверждения) для сеансов, которые должны быть завершены.

6) AS1 отвечает, передавая результат выхода, на запрос функции конечного пользователя, который был сделан на шаге 1.

6.10.2 Пользователь указывает непосредственно поставщику IdSP, что он/она желает выйти из всех сеансов

На рисунке 12 показаны основные шаги потока сообщений, которые описаны ниже.

6.10.2.1 Объекты, участвующие в процедуре, и информационный поток

Объекты, участвующие в процедуре выхода, те же, что и описанные в пункте 6.10.1.1. В этом случае функция конечного пользователя имеет отдельный сеанс (сеанс 3) с A-2 (IdSP). Она использует этот сеанс для направления запроса единого выхода на шаге 1. Остальные шаги процедуры аналогичны шагам, описанным в пункте 6.10.1.1.

Рисунок 12 – Единый выход на основе SAML, запрошенный пользователем в IdSP

Основные шаги процедуры единого выхода:

1) Функция конечного пользователя вызывает запрос единого выхода непосредственно на A-2.

2) A-2 направляет запрос выхода <LogoutRequest> на AS1. Запрос должен быть подписан для аутентификации и защиты целостности согласно описанию в пункте 8.2.7 [ITU-T X.1141].

   После валидации запроса AS1 делает попытку завершить сеанс 1. Для этого AS1 аннулирует регистрационные данные аутентификации для данного сеанса (например, утверждения, cookie-файлы), что заставит функцию конечного пользователя пройти процедуру аутентификации, если она направит другой запрос на AS1.

3) A-2 направляет <LogoutRequest> на AS2 (он также направляет этот запрос на все серверы участвующих сеансов). Этот шаг аналогичен шагу 2.

   После валидации запроса выхода AS2 делает попытку завершить сеанс 2.

4) AS1 сообщает отправителю запроса на выход (A-2) результат попытки выхода, направляя сообщение <LogoutResponse>, которое должно быть подписано.

5) Аналогично действиям на предыдущем шаге AS2 сообщает отправителю запроса на выход (A-2) результат попытки выхода посредством подписанного сообщения <LogoutResponse>.

   После этого шага A-2 обновляет свой список активных сеансов и аннулирует регистрационные данные аутентификации (например, cookie-файлы, утверждения) для сеансов, которые должны быть завершены.

После валидации всех ответов на запрос выхода A-2 сообщает функции конечного пользователя результат единого выхода. Это ответ на запрос функции конечного пользователя, который был сделан на шаге 1.

7 Безопасность

Механизмы, охваченные в настоящей Рекомендации, а также механизмы, определенные в [ITU-T Y.2704], отвечают требованиям к безопасности IdM, определенным в [ITU-T Y.2721].

Дополнение I

Аутентификация сообщения WSS МСЭ-T X.509 v3

(Это Дополнение не является неотъемлемой частью настоящей Рекомендации.)

Представленный ниже пример иллюстрирует способ обработки сообщений с жетоном WSS МСЭ-T X.509, описанный в пункте 6.5.1.2.

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

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

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

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

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

<wsse:Security mustUnderstand="1">

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

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

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

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

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

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

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

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

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

```
                        <ds:DigestValue>Ru4cAfeBABE...</ ds:DigestValue>
                  </ds:Reference>

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

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


</s:Envelope>
```

Дополнение II

Механизм управления доступом на базе "OpenID + OAuth"

(Это Дополнение не является неотъемлемой частью настоящей Рекомендации.)

В пункте 6.2.8 описано использование OpenID для аутентификации пользователя для любой сетевой прикладной функции (NAF). Таким образом, предлагается ввести OAuth в OpenID для защиты PII управления доступом.

II.1 OAuth [b-IETF RFC 5849]

OAuth – это открытый протокол, позволяющий приложению получить доступ к информации конечного пользователя из веб-услуги, при условии что это приложение авторизовано этим конечным пользователем. Информация конечного пользователя передается безопасным образом без раскрытия идентичности пользователя.

Целью OAuth является получение жетона доступа от веб-сервера, который может далее использоваться для обмена пользовательскими данными с этой веб-услугой (такими как расписание или адресная книга). Обычным процессом OAuth является состоящая из четырех шагов последовательность:

1) обращение за жетоном "запрос";
2) обращение за авторизацией жетона, что инициирует утверждение со стороны пользователя;
3) обмен авторизованного жетона запроса на жетон "доступ";
4) использование жетона доступа для взаимодействия с пользовательскими данными веб-услуги.

Более подробное описание OAuth содержится в [b-IETF RFC 5849].

II.2 Использование OpenID совместно с OAuth

В то время как OpenID может использоваться в качестве механизма IdM для аутентификации пользователей, OAuth в свою очередь может также использоваться для авторизации разрешения в отношении конфиденциальных пользовательских данных. При таком сценарии IdSP предоставляет объединенные функции и служит как поставщиком данных идентичности OpenID (OP), так и поставщиком услуг OAuth.

II.3 Поток сообщений авторизации OpenID + OAuth

В случае OpenID + OAuth эта последовательность остается практически такой же. Различие заключается в том, что получение авторизованного жетона запроса OAuth (шаги 1 и 2) запаковано в запрос аутентификации OpenID. Таким образом, пользователь может утвердить регистрацию и доступ к услуге одновременно.

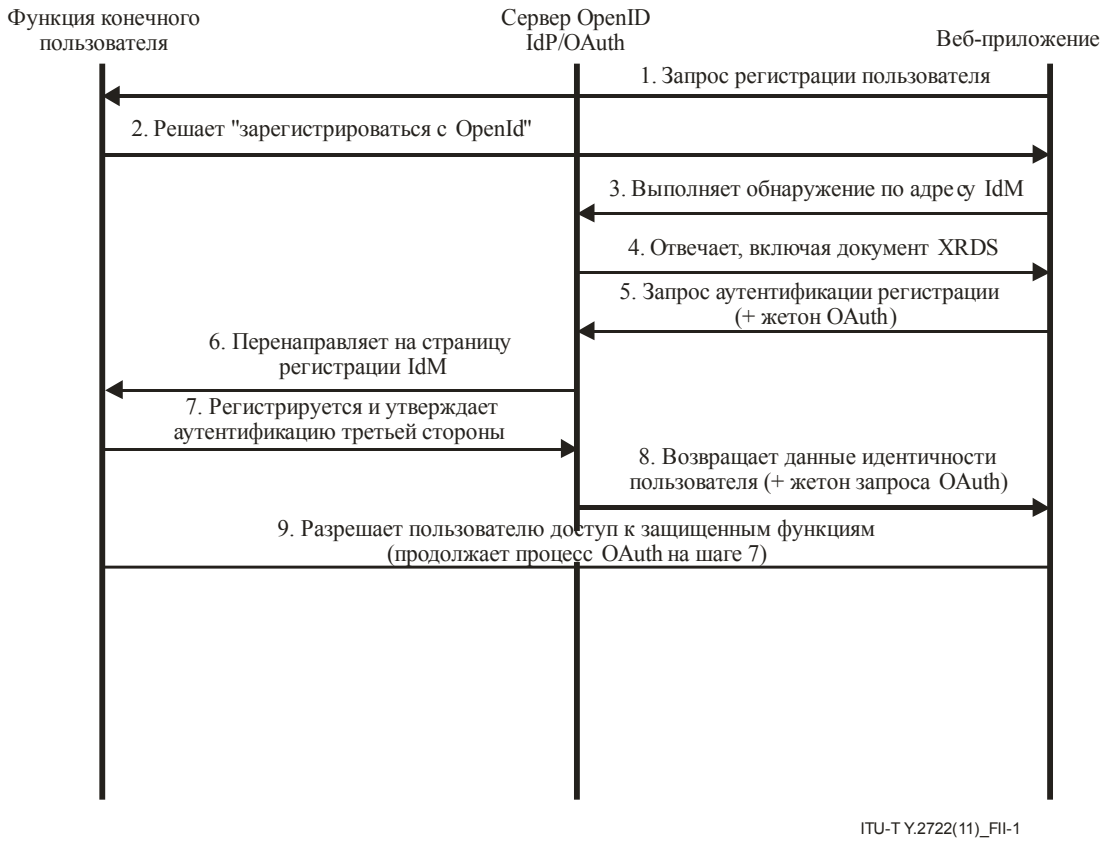


Рисунок II.1 – Аутентификация на базе OpenID + OAuth

Ниже описаны основные шаги:

1) Веб-приложение просит конечного пользователя зарегистрироваться, предлагая набор вариантов регистрации, включая использование его счета OpenID.

2) Пользователь выбирает вариант "Регистрация с OpenID".

3) Веб-приложение направляет запрос "обнаружение" поставщику IdSP для получения информации о конечной точке аутентификации регистрации IdSP.

4) IdSP возвращает документ XRDS, который содержит адрес конечной точки.

5) Веб-приложение направляет запрос аутентификации регистрации в адрес конечной точки IdSP.

6) Это действие перенаправляет пользователя на страницу федеративной регистрации IdSP либо в том же окне навигации, либо во всплывающем окне, и пользователю предлагается выполнить регистрацию.

7) После того как пользователь зарегистрировался, IdSP отображает страницу подтверждения и извещает пользователя о том, что приложение третьей стороны запрашивает аутентификацию. На этой странице пользователю предлагается подтвердить или отклонить связывание его регистрации для счета IdSP с регистрацией для веб-приложения. Далее пользователю предлагается утвердить доступ к конкретному набору услуг IdSP. Совместное использование регистрационной и пользовательской информации должно утверждаться пользователем для продолжения аутентификации.

8) Если пользователь утверждает аутентификацию, IdSP возвращает пользователя на URL, определенный в параметре `openid.return_to` исходного запроса. Представленный IdSP идентификатор, который не связан с фактическими именем и паролем счета IdM пользователя, добавляется как параметр запроса `openid.claimed_id`. Если запрос также включает обмен атрибутами, может быть добавлена дополнительная информация пользователя. В случае OpenID + OAuth также возвращается авторизованный жетон запроса OAuth.

9) Веб-приложение использует представленный IdSP идентификатор для узнавания пользователя и разрешения доступа к функциям и данным приложения. В случае OpenID + OAuth веб-приложение использует жетон запроса для продолжения последовательности OAuth и получения доступа к услугам IdSP пользователя.

Библиография

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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 протокола Интернет и сети последующих поколений**
Серия Z	Языки и общие аспекты программного обеспечения для систем электросвязи

Отпечатано в Швейцарии
Женева, 2012 г.