МСЭ-Т H.235

СЕКТОР СТАНДАРТИЗАЦИИ
ЭЛЕКТРОСВЯЗИ МСЭ

(08/2003)

СЕРИЯ H: АУДИОВИЗУАЛЬНЫЕ И МУЛЬТИМЕДИЙНЫЕ СИСТЕМЫ

Инфраструктура аудиовизуальных служб – Системные аспекты

Средства защиты и шифрования для мультимедийных терминалов серии H (терминалов H.323 и других терминалов на основе H.1245)

Рекомендация МСЭ-Т H.235

РЕКОМЕНДАЦИИ МСЭ-Т СЕРИИ H

АУДИОВИЗУАЛЬНЫЕ И МУЛЬТИМЕДИЙНЫЕ СИСТЕМЫ

ХАРАКТЕРИСТИКИ ВИДЕОТЕЛЕФОННЫХ СИСТЕМ	H.100–H.199
ИНФРАСТРУКТУРА АУДИОВИЗУАЛЬНЫХ СЛУЖБ	
Общие положения	H.200–H.219
Мультиплексирование и синхронизация при передаче	H.220–H.229
Системные аспекты	**H.230–H.239**
Процедуры связи	H.240–H.259
Кодирование подвижных видеоизображений	H.260–H.279
Соответствующие системные аспекты	H.280–H.299
СИСТЕМЫ И ОКОНЕЧНОЕ ОБОРУДОВАНИЕ ДЛЯ АУДИОВИЗУАЛЬНЫХ СЛУЖБ	H.300–H.399
ДОПОЛНИТЕЛЬНЫЕ УСЛУГИ ДЛЯ МУЛЬТИМЕДИЙНЫХ СЛУЖБ	H.450–H.499
ПРОЦЕДУРЫ МОБИЛЬНОСТИ И СОВМЕСТНОЙ РАБОТЫ	
Обзор мобильности и совместной работы, определений, протоколов и процедур	H.500–H.509
Мобильность для мультимедийных систем и служб серии H	H.510–H.519
Приложения и службы мобильной мультимедийной совместной работы	H.520–H.529
Безопасность для мобильных мультимедийных систем и служб	H.530–H.539
Безопасность для приложений и служб мобильной мультимедийной совместной работы	H.540–H.549
Процедуры мобильного взаимодействия	H.550–H.559
Процедуры взаимодействия мобильной мультимедийной совместной работы	H.560–H.569
ШИРОКОПОЛОСНЫЕ МУЛЬТИМЕДИЙНЫЕ СЛУЖБЫ И МУЛЬТИМЕДИЙНЫЕ СЛУЖБЫ В РЕЖИМЕ TRIPLE-PLAY	
Предоставление широкополосных мультимедийных услуг по VDSL	H.610–H.619

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

Рекомендация МСЭ-Т H.235

Средства защиты и шифрования для мультимедийных терминалов серии H (терминалов H.323 и других терминалов на основе H.245)

Резюме

В настоящей Рекомендации описываются усовершенствования, внесенные в рамках Рекомендаций серии H.3xx в части включения таких сетевых средств защиты, как *аутентификация* и *секретность* (шифрование данных). Предлагаемая схема применима как в простых двусторонних конференциях, так и в многосторонних, для любых терминалов, использующих протокол Рек. МСЭ-Т H.245 в качестве протокола управления.

Например, системы H.323 функционируют в сетях, основывающихся на пакетной передаче, которые не обеспечивают гарантированное качество обслуживания. По тем же техническим причинам, по которым базовая сеть не гарантирует соответствующее качество обслуживания (QoS), эта сеть не обеспечивает сетевые средства защиты. Защищенная передача информации в реальном времени по незащищенным сетям обычно имеет две основные проблемные области – *аутентификацию и секретность*.

В этой Рекомендации описаны инфраструктура защиты и конкретные средства защиты, которые следует применять в мультимедийных терминалах серии H.3xx. Настоящая Рекомендация охватывает проблемы проведения интерактивных конференций. Они включают аутентификацию и секретность всех потоков мультимедийной информации реального времени, обмен которыми производится в ходе конференции (но не ограничиваются только этим). В настоящей Рекомендации представлены протокол и алгоритмы, необходимые при взаимодействии объектов H.323.

В данной Рекомендации используются те же стандартные функции, что представлены в Рек. МСЭ-Т H.245, и таким образом, любое стандартное устройство, функционирующее во взаимодействии с управляющим протоколом, может использовать эту схему защиты. Подразумевается, что там, где это возможно, иные терминалы серии H могут взаимодействовать и непосредственно использовать методы, описанные в данной Рекомендации. Настоящая Рекомендация в сущности не претендует на описание всех областей реализации, но будет касаться конкретно аутентификации конечных точек и секретности передаваемой мультимедийной информации.

Настоящая Рекомендация включает возможность общего согласования сетевых средств и функциональных возможностей, а также выбора применяемых криптографических методов и средств. Конкретный способ их использования зависит от возможностей системы, требований реализации и ограничений, обусловленных конкретной стратегией защиты. В настоящей Рекомендации представлены различные криптографические алгоритмы, разные опции которых (например, длины ключей) предусмотрены для различных целей. Определенные криптографические алгоритмы могут распределяться конкретным сетевым средствам защиты (например, один – для быстрого шифрования потоков мультимедийной информации, а другой – для шифрования сигнальной информации).

Следует отметить, что некоторые из имеющихся криптографических алгоритмов или механизмов могут быть зарезервированы для экспорта информации или иных проблем национального уровня (например, при ограниченных длинах ключей). Настоящая Рекомендация поддерживает сигнализацию хорошо известных алгоритмов, а также нестандартизованных или частных криптографических алгоритмов. Не существует специально предписанных алгоритмов, однако, настоятельно рекомендуется, чтобы конечные точки поддерживали как можно больше приемлемых алгоритмов для достижения взаимодействия. Это соответствует концепции, согласно которой реализация Рек. МСЭ-Т H.245 не гарантирует взаимодействие между кодеками двух объектов.

Версия 2 Рек. МСЭ-Т H.235 заменяет версию 1 Рек. МСЭ-Т H.235 и включает ряд усовершенствований, в том числе, шифрование методом эллиптических кривых, профили защиты (простые, основанные на пароле, и сложные – в виде цифровой подписи), новые меры противодействия при нарушении защиты (средства защиты от спама мультимедийной информации), поддержку Усовершенствованного стандарта шифрования (AES), поддержку внутренних серверов, определение идентификаторов объектов и изменения, внесенные из руководства по реализации стандарта H.323.

Рек. МСЭ-Т H.235 версия 3 служит заменой Рек. МСЭ-Т H.235 версии 2 и включает процедуру для шифрованных двухтональных многочастотных (DTMF) сигналов, идентификаторы объектов для шифрования мультимедийной полезной нагрузки при использовании алгоритма шифрования AES, усовершенствованный метод шифрования потоков информации с обратной связью по выходу (EOFB) для шифрования потоков мультимедийной информации, опцию, рассчитанную только на аутентификацию, представленную в Приложении D, для беспрепятственной трансляции сетевых адресов (NAT)/прохождения брандмауэров (firewall), а также процедуру распределения ключей по каналам регистрации, доступа и статуса (RAS), процедуры для более безопасной передачи сеансовых ключей и более надежного распределения и обновления сеансовых ключей, процедуры для организации защиты множественных потоков полезной нагрузки, более совершенные средства защиты прямо маршрутизируемых вызовов, которые представлены в новом Приложении I, средства сигнализации для организации более гибкого сообщения об ошибке, средства и способы повышения эффективности защиты информации при быстром старте и при сигнализации Диффи-Хеллмана (Diffie-Hellman), а также – параметры Диффи-Хеллмана и изменения, внесенные из руководства по реализации, данного в Рек. H. 323 .

Источник

Рекомендация МСЭ-Т H.235 была утверждена Исследовательской комиссией 16 МСЭ-Т (2001–2004 гг.) по процедуре, представленной в Рекомендации МСЭ-Т A.8, 6 августа 2003 года.

Ключевые слова

Аутентификация, сертификат, цифровая подпись, шифрование, целостность, управление ключами, защита мультимедийной информации, профиль защиты.

История

Версия	Утверждение
H.235v1	1998-02-06
H.235v2	2000-11-17
H.235v3	2003-08-06

ПРЕДИСЛОВИЕ

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

Всемирная ассамблея по стандартизации электросвязи (ВАСЭ), которая проводится каждые четыре года, определяет 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 технологии, 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.

Соответствие положениям данной Рекомендации является добровольным делом. Однако в Рекомендации могут содержаться определенные обязательные положения (для обеспечения, например, возможности взаимодействия или применимости), тогда соответствие данной Рекомендации достигается в том случае, если выполняются все эти обязательные положения. Для выражения требований используются слова "shall" ("должен", "обязан") или некоторые другие обязывающие термины, такие как "must" ("должен"), а также их отрицательные эквиваленты. Использование таких слов не предполагает, что соответствие данной Рекомендации требуется от каждой стороны.

ПРАВА ИНТЕЛЛЕКТУАЛЬНОЙ СОБСТВЕННОСТИ

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

На момент утверждения настоящей Рекомендации МСЭ получил извещение об интеллектуальной собственности, защищенной патентами, которые могут потребоваться для реализации этой Рекомендации. Однако те, кто будет применять Рекомендацию, должны иметь ввиду, что это может не отражать самую последнюю информацию, и поэтому им настоятельно рекомендуется обращаться к патентной базе данных БСЭ.

СОДЕРЖАНИЕ

Стр.

Стр.

Рекомендация МСЭ-Т H.235

Средства защиты и шифрования для мультимедийных терминалов серии H (терминалов H.323 и других терминалов на основе H.245)

1 Область рассмотрения

Настоящая Рекомендация посвящена, главным образом, вопросам аутентификации, обеспечению секретности и целостности информации в рамках существующих протоколов серии H. В тексте данной Рекомендации (2003 г.) содержатся подробности реализации положений Рекомендации МСЭ-Т H.323. Предполагается, что данный протокол будет взаимодействовать с другими протоколами серии H, использующими в качестве своего протокола управления тот, который описан в Рек. МСЭ-Т H.245.

К дополнительным целям настоящей Рекомендации относятся следующие:

1) Разработка архитектуры защиты в виде расширяемой и гибкой структуры для реализации системы защиты для терминалов серии H. Эта цель может быть достигнута посредством предоставления гибких и независимых сетевых средств и обеспечиваемых ими функциональных возможностей. Сюда относится способность согласования и выбора используемых криптографических методов, а также способ их использования.

2) Обеспечение защиты всех соединений, устанавливающихся в результате применения протокола H.3xx. Это включает установление соединения, управление вызовом и обмен мультимедийной информацией между всеми объектами. Настоящее требование включает обеспечение конфиденциальности связи (секретности); для этого может применяться аутентификация одноранговых объектов, а также защита операционной среды пользователя от попыток ее нарушения.

3) Настоящая Рекомендация не должна препятствовать интеграции иных функций защиты в объекты H.3xx, которые могут противостоять попыткам нарушения защиты со стороны сети.

4) Настоящая Рекомендация не должна ограничивать возможность соответствующих изменений любой Рекомендации серии H.3xx. Это может касаться как количества защищенных пользователей, так и обеспечиваемых уровней защиты.

5) По мере возможности, все механизмы и устройства должны предоставляться независимо от используемого транспорта или топологий сети. Для преодоления этого могут потребоваться иные средства, выходящие за рамки настоящей Рекомендации.

6) Должны быть разработаны положения по работе в смешанной операционной среде (содержащей защищенные и незащищенные объекты).

7) Настоящая Рекомендация должна обеспечить средства для распределения сеансовых ключей, соответствующих используемому методу криптографии. (Это не предполагает, что вопрос управления сертификатами с открытым ключом должен быть частью настоящей Рекомендации.)

8) В данной Рекомендации предусмотрены два профиля защиты, повышающие возможность взаимодействия. В Приложении D описывается простой, но тем не менее надежный профиль защиты, основывающийся на пароле, а Приложение E посвящено профилю защиты в виде цифровых подписей, сертификатов и инфраструктуры шифрования с открытым ключом, что выходит за рамки вопросов, рассматриваемых в Приложении D.

Архитектура защиты, описанная в настоящей Рекомендации, не предполагает, что участники конференции знакомы друг с другом. Однако, предполагается, что были предприняты необходимые меры предосторожности для того, чтобы физически защитить конечные точки из Рекомендаций серии H. Следовательно, ожидается, что основная угроза защите информации при передаче – это перехват информации или какой-либо другой способ нарушения потоков мультимедийной информации.

В Рек. МСЭ-Т H.323 представлены средства и способы проведения аудиоконференций, видеоконференций и конференций с передачей данных между двумя или более участниками, однако, не указан ни механизм, позволяющий каждому участнику идентифицировать других участников, ни средства, позволяющие сделать связь секретной (т. е. зашифровать потоки).

В Рек. МСЭ-Т H.323, H.324 и H.310 используются процедуры сигнализации по логическим каналам Рек. МСЭ-Т H.245, в которых описывается содержимое каждого логического канала, когда канал открыт. Предусмотрены процедуры для того, чтобы показать возможности получателя и отправителя, так как передача ограничена возможностями декодирования со стороны получателей, и получатели могут потребовать от отправителей определенного режима работы. О возможностях обеспечения защиты информации каждой конечной точки сообщается таким же образом, как и о любых других возможностях средств связи.

Некоторые терминалы серии H (H.323) могут применяться в многоточечных конфигурациях. Механизм защиты, описанный в настоящей Рекомендации, позволит обеспечить безопасное функционирование в условиях данной операционной среды, включая как централизованное, так и децентрализованное функционирование MCU.

2 Источники ссылок

Следующие рекомендации МСЭ-Т и другие источники содержат положения, которые, будучи упомянутыми в качестве ссылок в данном тексте, составляют положения данной Рекомендации. На момент публикации указанные издания были действующими. Все рекомендации и другие источники ссылок подлежат пересмотру; поэтому всем пользователям этих рекомендаций предлагается рассмотреть возможность использования самого последнего издания этих рекомендаций и других источников ссылок, перечисленных ниже. Перечень действующих рекомендаций МСЭ-Т публикуется регулярно. Ссылка на любой документ в рамках данной Рекомендации не придает ему, даже при том, что это отдельный документ, статуса рекомендации.

– Рекомендация МСЭ-Т H.225.0 (2003 г.), *Протоколы передачи сигналов вызова и пакетирование потоков мультимедийной информации для мультимедийных систем связи в пакетном режиме.*

– Рекомендация МСЭ-Т H.235 (1998 г.), *Средства защиты и шифрования для мультимедийных терминалов серии H (терминалов H.323 и других на основе H.245).*

– Рекомендация МСЭ-Т H.235 (2000 г.), *Средства защиты и шифрования для мультимедийных терминалов серии H (терминалов H.323 и других на основе H.245).*

– Рекомендация МСЭ-Т H.530 (2002 г.), *Процедуры симметричной защиты информации при условии мобильности пользователей H.323 в H.510.*

– Рекомендация МСЭ-Т H.530 Поправка 1 (2003 г.), *Процедуры симметричной защиты информации при условии мобильности пользователей H.323 в H.510.*

– Рекомендация МСЭ-Т H.245 (2003 г.), *Протокол управления для мультимедийной связи.*

– Рекомендация МСЭ-Т H.323 (2003 г.), *Мультимедийные системы связи в пакетном режиме.*

– Рекомендация МСЭ-Т Q.931 (1998 г.), *Спецификации уровня 3 интерфейса пользователь–сеть ЦСИС для управления основным вызовом.*

– Рекомендация МСЭ-Т X.509 (2000 г.) | ИСО/МЭК 9594-8:2001, *Информационные технологии – Взаимосвязь открытых систем – Справочник: Структуры открытых ключей и прилагаемых сертификатов.*

– Рекомендация МСЭ-Т X.800 (1991 г.), *Архитектура защиты для взаимосвязи открытых систем для приложений МККТТ.*

– ИСО 7498-2:1989, *Системы обработки информации – Взаимосвязь открытых систем – Базовая эталонная модель – Часть 2: Архитектура защиты.*

– Рекомендация МСЭ-Т X.803 (1994 г.) | ИСО/МЭК 10745:1995, *Информационные технологии – Взаимосвязь открытых систем – Модель защиты верхних уровней.*

– Рекомендация МСЭ-Т X.810 (1995 г.) | ИСО/МЭК 10181-1:1996, *Информационные технологии – Взаимосвязь открытых систем – Схемы защиты для открытых систем: Обзор.*

– Рекомендация МСЭ-Т X.811 (1995 г.) | ИСО/МЭК 10181-2:1996, *Информационные технологии – Взаимосвязь открытых систем – Схемы защиты для открытых систем: Схемы аутентификации.*

– ИСО/МЭК 9797:1994, *Информационные технологии – Методы защиты – Механизм обеспечения целостности данных, использующий функцию криптографической проверки с применением алгоритма блочного шифрования.*

– ИСО/МЭК 9798-2:1999, *Информационные технологии – Методы защиты – Аутентификация объектов – Часть 2: Механизмы, использующие алгоритмы симметричного шифрования.*

– ИСО/МЭК 9798-3:1998, *Информационные технологии – Методы защиты – Аутентификация объектов – Часть 3: Механизм, использующий методы цифровой подписи.*

– ИСО/МЭК 9798-4:1999, *Информационные технологии – Методы защиты – Аутентификация объектов – Часть 4: Механизмы, использующие функцию криптографической проверки.*

– ИСО/МЭК 10116:1997, *Информационные технологии – Методы защиты – Методы защиты n-битового блочного шифра.*

– ИСО/МЭК 15946-1:2002, *Информационные технологии – Методы защиты – Методы шифрования, основанные на эллиптических кривых – Часть 1: Общие сведения.*

– ИСО/МЭК 15946-2:2002, *Информационные технологии – Методы защиты – Методы шифрования, основанные на эллиптических кривых – Часть 2: Цифровые подписи.*

– Форум ATM: af-sec-0100.002 (2001 г.), *Спецификации защиты в режиме асинхронной передачи (ATM), Версия 1.1.*

– Стандарт IETF 1321 (1992), *Алгоритм передачи дайджестов сообщений MD5.*

– Стандарт IETF 2104 (1997 г.), *HMAC: Хеширование ключа для аутентификации сообщений.*

– Стандарт IETF 2865 (2000 г.), *Служба аутентификации удаленных пользователей по телефонным линиям (RADIUS).*

– Стандарт IETF 2198 (1997 г.), *Полезная нагрузка RTP для избыточных аудиоданных.*

– Стандарт IETF 2246 (1999 г.), *Протокол TLS, Версия 1.0.*

– Стандарт IETF 2401 (1998 г.), *Архитектура защиты для Интернет-протокола.*

– Стандарт IETF 2402 (1998 г.), *Заголовок сообщения аутентификации в протоколе IP.*

– Стандарт IETF 2407 (1998 г.), *Домен защиты интерпретации протокола IP для ISAKMP.*

– Стандарт IETF 2412 (1998 г.), *Протокол определения ключа OAKLEY.*

– Стандарт IETF 2437 (1998 г.), *PKCS #1: Спецификации системы шифрования RSA. Версия 2.0.*

– Стандарт IETF 2833 (2000 г.), *Полезная нагрузка RTP для цифровых сигналов DTMF, тональных и телефонных сигналов.*

– Стандарт IETF 3280 (2002 г.), *Сертификат инфраструктуры открытого ключа сети Интернет X.509 и профиль Перечня отмененных сертификатов (CRL).*

3 Термины и определения

На настоящую Рекомендацию, помимо определений, приведенных в настоящем пункте, распространяются и те, которые были даны в пунктах 3/H.323, 3/H.225.0 и 3/H.245. Некоторые из следующих терминов употребляются в соответствии с определениями в Рек. МСЭ-Т X.800 │ИСО 7498-2 и МСЭ-Т X.803, X.810 и X.811.

3.1 управление доступом (access control): Предотвращение несанкционированного использования какого-либо ресурса, в том числе предотвращение использования ресурса в режиме несанкционированных действий (Рек. МСЭ-Т X.800).

3.2 аутентификация (authentication): Обеспечение гарантированной и обусловленной тождественности объекта (Рек. МСЭ-Т X.811).

3.3 авторизация (authorization): Предоставление права на доступ на основании подтверждения права на доступ после аутентификации.

3.4 попытка нарушения защиты (attack): Действия, предпринятые для обхода или для использования недостатков в механизмах защиты системы. При прямой попытке нарушения защиты системы используются недостатки в базовых алгоритмах, принципах или свойствах защитного механизма. При косвенных попытках нарушения защиты осуществляется обход этого механизма или же система принуждается к неправильному применению этого механизма.

3.5 сертификат (certificate): Набор касающихся вопросов защиты данных, предоставляемый уполномоченным органом по вопросам защиты или доверенной третьей стороной, вместе с информацией о защите, используемой для обеспечения данными сетевых средств контроля целостности и аутентификации происхождения данных (Рек. МСЭ-Т X.810). В настоящей Рекомендации указанный термин касается сертификатов "открытого ключа", являющимся значениями, которые представляют собой открытый ключ владельцев полномочий (и иную дополнительную информацию), проверенный и удостоверенный доверенным органом в формате, который невозможно подделать.

3.6 кодирование (cipher): Криптографический алгоритм, математическое преобразование.

3.7 конфиденциальность (confidentiality): Свойство, которое предупреждает раскрытие информации неправомочным лицам, объектам или процессам.

3.8 криптографический алгоритм (cryptographic algorithm): Математическая функция, вычисляющая результат по одной или нескольким входным значениям.

3.9 шифрование (encipherment): Шифрование (засекречивание) – это процесс, преобразования данных в недоступную для прочтения неправомочными пользователями форму; для этого применяется криптографический алгоритм (алгоритм шифрования). Дешифрование – это обратная операция, при которой зашифрованный текст преобразуется в открытый текст.

3.10 целостность (integrity): Свойство, позволяющее сохранять данные неизменными в режиме несанкционированных действий.

3.11 управление ключами (key management): Создание, хранение, распределение, уничтожение, архивирование и применение ключей в соответствии со стратегией защиты (Рек. МСЭ-Т X.800).

3.12 поток мультимедийной информации (media stream): Поток мультимедийной информации может состоять из аудиосигналов, видеосигналов, данных или любой комбинации этих типов. Потоки мультимедийной информации переносят пользовательскую или прикладную информацию (полезную нагрузку), но не данные управления.

3.13 защита от неподтверждения (non-repudiation): Предупреждение отрицания факта участия во всем процессе связи или в его части со стороны одного из объектов, участвующих в этом процессе связи.

3.14 секретность (privacy): Режим связи, при котором интерпретировать передаваемые сообщения могут только стороны, получившие на это формальное разрешение. Обычно секретность достигается посредством шифрования и общего ключа (ключей) к шифру.

3.15 выделенный канал (private channel): В настоящей Рекомендации под выделенным каналом понимается канал, выделенный в результате предварительного согласования вопроса о защищенном канале. В данном контексте он может применяться для манипулирования потоками мультимедийной информации.

3.16 шифрование с открытым ключом (public key cryptography): Система шифрования, использующая асимметричные ключи (для шифрования/дешифрования), при которой между этими ключами существует математическая зависимость, которую нельзя определить с приемлемой степенью надежности.

3.17 профиль защиты (security profile): (Под)множество согласующихся взаимодействующих процедур и свойств из Рек. МСЭ-Т H.235, подходящих для защиты мультимедийной связи (H.323) между участвующими в связи объектами в рамках конкретного сценария.

3.18 спамминг (spamming): Попытка нарушения защиты типа "отказ в обслуживании" в виде отправки системе несанкционированных данных в избыточном количестве. Особым случаем является мультимедийный спамминг в виде отправки пакетов в RTP к портам в протоколе пользовательских дейтограмм (UDP). Обычно система переполнена пакетами, обработка которых задействует ценные системные ресурсы.

3.19 симметричный (основанный на секретном ключе) криптографический алгоритм (symmetric (secret-key based) cryptographic algorithm): Алгоритм для выполнения шифрования или соответствующий алгоритм для выполнения дешифрования, при котором для шифрования и дешифрования требуется один и тот же ключ (Рек. МСЭ-Т X.810).

3.20 угроза безопасности системы защиты данных (threat): Потенциальное нарушение защиты (Рек. МСЭ-Т X.800).

4 Символы и аббревиатуры

В настоящей Рекомендации используются следующие аббревиатуры:

X || Y	Конкатенация X и Y
3DES	Тройной DES
AES	Усовершенствованный стандарт шифрования
ASN.1	Абстрактно-синтаксическая нотация, версии 1
BES	Внутренний сервер
CA	Сертификационный центр
CBC	Сцепление шифрованных блоков
CFB	Режим обратной связи по шифрованному тексту
CRL	Перечень отмененных сертификатов
DES	Стандарт шифрования данных
DH	Алгоритм Диффи-Хеллмана
DNS	Служба имен доменов
DSS	Стандарт цифровой подписи
DTMF	Двухтональный многочастотный
ECB	Электронная кодовая книга
ECC и EC	Система шифрования методом эллиптических кривых (см. раздел 8.7 *"Форум АТМ. Спецификация защиты", Версия 1.1*), Система шифрования открытым ключом
EC-GDSA	Цифровая подпись, шифрованная методом эллиптических кривых с дополнительным аналогом Алгоритма шифрования цифровой подписи (DSA) Национального института стандартов и технологий (NIST) (см. также ИСО/МЭК 15946-2, глава 5)
ECKAS-DH	Схема согласования ключей при шифровании методом эллиптических кривых – Диффи-Хеллмана (Diffie-Hellman). Схема согласования ключей Диффи-Хеллмана, использующая шифрование методом эллиптических кривых
EOFB	Усовершенствованный режим OFB
EP	Конечная точка
GK	Контроллер доступа
GW	Шлюз
ICV	Контрольный признак целостности
ID	Идентификатор
IPSEC	Защита на уровне Интернет-протокола
ISAKMP	Протокол управления ключами и ассоциация защиты сети Интернет
IV	Вектор инициализации
LDAP	Упрощенный протокол доступа к сетевым каталогам
MAC	Код аутентификации сообщений
MCU	Устройство управления многосторонней связью
MD5	Дайжест (сообщение) MD5
MPS	Групповой поток полезной нагрузки

NAT	Трансляция сетевых адресов
OCSP	Протокол оперативного состояния сертификата
OFB	Режим обратной связи (с выхода)
OID	Идентификатор объекта
PDU	Протокольный блок данных
PKCS	Система шифрования с открытым ключом
PKI	Инфраструктура открытого ключа
PRF	Псевдослучайная функция
QoS	Качество обслуживания
RSA	Алгоритм Райвеста-Шамира-Адельмана (алгоритм шифрования с открытым ключом)
RTCP	Протокол управления передачей данных в режиме реального времени
RTP	Протокол передачи данных в режиме реального времени
SDU	Сервисный блок данных
SHA1	Алгоритм аутентификации и проверки целостности информации 1
SRTP	Протокол защищенной передачи данных в режиме реального времени
SSL	Уровень защитных идентифицирующих параметров (сокетов)
TLS	Протокол обеспечения защиты транспортного уровня
TSAP	Точка доступа к услугам транспортного уровня
XOR, ⊕	Исключающее ИЛИ

5 Принятые понятия и условные обозначения

В настоящей Рекомендации следующие слова понимаются указанным ниже образом:

– "должен" означает обязательное требование;
– "должен или следует" обозначает предлагаемый, но необязательный образ действий;
– "может или возможно" означает скорее необязательный образ действий, чем рекомендацию что-либо предпринять.

Ссылки на пункты, подпункты, приложения и дополнения относятся к соответствующим позициям в рамках настоящей Рекомендации, если нет явного указания на другую рекомендацию. Например, "1.4" относится к пункту 1.4 настоящей Рекомендации; "6.4/H.245" относится к пункту 6.4 в Рекомендации H.245.

В настоящей Рекомендации описано применение "n" различных типов сообщений: H.245, RAS, Q.931 и т. д. Для того, чтобы можно было отличить друг от друга сообщения различных типов, принято следующее. Названия параметров и сообщений H.245 состоят из множества "сцепленных" слов, выделенных жирным шрифтом (**maximumDelayJitter**). Названия сообщений RAS представлены в виде сокращений из трех букв (**ARQ**). Названия сообщений Q.931 состоят из одного или двух слов, начинающихся с заглавных букв (**Call Proceeding**).

В этой Рекомендации определены различные идентификаторы объектов (OID) для возможностей защиты при передаче сигналов, процедур или алгоритмов защиты. Эти OID соотносятся с иерархическим деревом присвоенных значений, корнем которых могут послужить внешние источники, или же они могут быть частью дерева OID, поддерживаемого Рек. МСЭ-Т. Те OID, которые конкретно относятся к Рек. МСЭ-Т H.235, имеют в тексте следующий вид:

"OID" = {itu-t (0) recommendation (0) h (8) 235 version (0) **V N**}, где **V** символически представляет одну десятичную цифру, обозначающую соответствующую версию Рек. МСЭ-Т H.235; к примеру, 1, 2, или 3.

N – символически представляет десятичное число, однозначно определяющее экземпляр OID и, таким образом, процедуру, алгоритм или возможности защиты.

Таким образом, закодированный в ASN.1 OID представляет собой последовательность чисел. Для удобства в тексте используется мнемоническая короткая строковая нотация для каждого OID, такая как "OID". Представлена схема распределения, которая соотносит каждую строку OID с последовательностью чисел в ASN.1. При реализациях, соответствующих Рек. МСЭ-Т H.235, необходимо использовать числа, кодированные только в ASN.1.

Что касается реализации шифрования мультимедийной информации в сочетании со вставкой информационных сигналов, то этот текст в некоторых местах гласит, что: "величина вставки должна обычно определяться в соответствии с алгоритмом шифрования", см., к примеру, 8.6.1, B.2.4 и рисунок I.5. Это означает, что некоторые алгоритмы шифрования (к примеру, DES) обеспечивают дополнительные практические сведения о том, каким образом отправитель может выбрать величину бита(ов) заполнения. Примерами могут служить произвольные величины заполнения, статические величины или другие генерируемые кодовые комбинации. Какой бы метод ни применялся, он не влияет на способность к взаимодействию, даже при значительном отличии уровня защиты. Этот вопрос касается проблем реализаций и не рассматривается подробно далее в этой Рекомендации.

6 Реализация системы

6.1 Резюме

1) Канал сигнализации вызова может быть защищен с помощью протоколов TLS [TLS] или IPSEC [IPSEC] в закрепленном защищенном порту (Рек. МСЭ-Т H.225.0).

2) Аутентификация пользователей может осуществляться во время первоначального установления соединения, в процессе обеспечения защиты канала H.245, и/или посредством обмена сертификатами по каналу H.245.

3) Возможности шифрования мультимедийного канала определяются исходя из расширения существующих возможностей механизма взаимодействия.

4) Первоначальное распределение данных ключей от ведущего терминала осуществляется посредством сообщений H.245 **OpenLogicalChannel** или **OpenLogicalChannelAck**.

5) Повторные манипуляции с ключами могут выполняться посредством команд H.245: **EncryptionUpdateCommand**, **EncryptionUpdateRequest**, **EncryptionUpdate и EncryptionUpdateAck**.

6) Для защиты при распределении данных ключа используется или канал H.245 в качестве выделенного канала, или специально обеспечивается защита данных ключей путем применения выбранных для обмена сертификатов.

7) Представленные протоколы защиты соответствуют или официальным стандартам ISO, или предложенным стандартам IETF.

6.2 Аутентификация

В ходе процедуры аутентификации происходит подтверждение, путем проверки, того, что респонденты на самом деле являются теми, за кого они себя выдают. Аутентификация может осуществляться совместно с обменом сертификатами, основанными на шифровании открытым ключом. Аутентификация может также выполняться путем обмена между участвующими объектами общим "ключом". Это может быть статический пароль или некоторый иной *заранее* согласованный элемент информации.

В настоящей Рекомендации описан протокол обмена сертификатами, но не определены критерии взаимной проверки и признания. В целом, сертификаты позволяют проверяющей стороне в некоторой степени убедиться в том, что предъявитель сертификата на самом деле является тем, за кого он себя выдает. Обмен сертификатами производится с целью аутентификации *пользователя* в конечной точке, а не просто – физической конечной точки. С помощью цифровых сертификатов протокол аутентификации проверяет, обладают ли респонденты личными ключами, соответствующими открытым ключам, содержащимся в сертификатах. Такая аутентификация обеспечивает защиту от попыток нарушения со стороны постороннего лица ("man-in-the-middle"), но не служит автоматическим доказательством того, кем являются респонденты. С этой целью обычно требуется наличие какой-либо стратегии в отношении остального содержимого сертификатов. Например, сертификаты авторизации обычно включают идентификатор поставщика услуг, а также некий идентификатор учета пользователей, устанавливаемый поставщиком услуг.

Схема аутентификации, описанная в настоящей Рекомендации, не определяет содержимое сертификатов (т. е. не определяют стратегию формирования сертификата) сверх того, что востребовано протоколом аутентификации. Однако, при применении этой схемы могут возникнуть требования стратегии более высокого уровня (например, необходимость представления сертификата пользователю для получения его одобрения). Такая стратегия более высокого уровня может быть или автоматизирована в рамках конкретного приложения, или же она может потребовать взаимодействия между людьми.

Для процедуры аутентификации без применения цифровых сертификатов в настоящей Рекомендации предусмотрены различные сценарии – от передачи сигналов до полной двусторонней аутентификации. При таком методе аутентификации требуется предварительная координация между объектами, участвующими в процессе связи, с тем, чтобы можно было получить общий "ключ". Примером применения этого метода может служить пользователь услуги, предоставляемой по подписке.

В третьем варианте, аутентификация может совершаться в рамках контекста отдельного протокола защиты, такого, как TLS [TLS] или IPSEC [IPSEC].

Аутентификация может поддерживаться одноранговыми объектами как в двух, так и в одном направлении. Аутентификация может выполняться по некоторым или по всем каналам связи.

Все конкретные механизмы аутентификации, описанные в настоящей Рекомендации, или идентичные разработанным ISO алгоритмам, описанным в Частях 2 и 3 документа ИСО/МЭК 9798, или получены исходя из этих алгоритмов, или базируются на протоколах IETF.

6.2.1 Сертификаты

Вопрос стандартизации сертификатов, включая их создание, администрирование и распределение, выходят за рамки настоящей Рекомендации. Сертификаты, используемые для создания защищенных каналов (передачи сигналов вызова и/или управление вызовом), должны соответствовать тем, которые предписываются любым протоколом, согласованным для обеспечения защиты этого канала.

Следует отметить, что при процедуре аутентификации с применением сертификатов открытых ключей требуется, чтобы конечные точки предоставляли цифровые подписи, используя соответствующее значение личного ключа. Сам по себе обмен сертификатами открытых ключей не защищает от попыток нарушения защиты со стороны посторонних лиц. Протоколы H.235 соответствуют этому требованию.

6.3 Защита при установлении соединения

Существуют, как минимум, две причины, обуславливающие необходимость защиты канала установления соединения (например, H.323 с применением Q.931). Первая – связана с возможностью простой аутентификации перед приемом вызова. Вторая – связана с обеспечением авторизации вызова. Если такая функция желательна для терминала серии H, то следует использовать режим защиты связи (например, TLS/IPSEC для H.323) прежде, чем обмениваться сообщениями об установлении соединения. В ином случае, авторизация может быть основана на аутентификации конкретной службы. Ограничения в части стратегии авторизации конкретной службы выходят за рамки настоящей Рекомендации.

6.4 Защита управления вызовом (H.245)

Канал управления вызовом (H.245) также следует некоторым образом защищать в целях обеспечения соответствующей секретности мультимедийной информации. Канал H.245 должен быть защищен посредством применения любого согласованного механизма обеспечения секретности (сюда относится и опция "none"). Сообщения H.245 используются для передачи алгоритмов шифрования сигналов и ключей шифрования, применяемых в выделенных, общих мультимедийных каналах. Возможность осуществления этого в логическом канале, используя базу логического канала, позволяет шифровать различные мультимедийные каналы посредством различных механизмов. Например, при централизованных многосторонних конференциях для потоков информации к каждой конечной точке могут использоваться разные ключи. Благодаря этому, потоки мультимедийной информации можно сделать индивидуальными для каждой конечной точки в конференции. Для того, чтобы защитить применение сообщений H.245, следует открыть весь канал H.245 (логический канал 0) в согласованном режиме защиты.

Механизм, посредством которого обеспечивается защита H.245, зависит от участвующих в процессе связи терминалов серии H. Единственное требование ко всем системам, использующим эту схему защиты, состоит в том, что для каждой из этих систем должен быть предусмотрен некий метод согласования и/или передачи сигналов о том, что канал H.245 должен работать в определенном режиме защиты до его

фактической инициализации. Например, для выполнения этого условия H.323 использует сигнальные сообщения H.225.0 об установлении соединения.

6.5 Секретность потоков мультимедийной информации

В настоящей Рекомендации рассматривается вопрос секретности потоков мультимедийной информации, передаваемых в пакетном режиме по транспортным каналам. Эти каналы могут быть однонаправленными с точки зрения характеристик логического канала H.245. Не требуется, чтобы указанные каналы были однонаправленными на физическом или транспортном уровне.

Первый шаг к достижению секретности мультимедийной информации должен заключаться в обеспечении выделенного канала управления, по которому можно было бы формировать данные о настройке по ключу в криптографии и/или организовывать логические каналы, по которым будут передаваться зашифрованные потоки мультимедийной информации. Для этого, участвуя в защищенной конференции конечные точки могут использовать зашифрованный канал H.245. При этом ключи шифрования и выбора криптографического алгоритма, передаваемые в команде H.245 **OpenLogicalChannel**, защищены.

Возможно использование защищенного канала H.245 с характеристиками, отличными от тех, которые имеет выделенный мультимедийный канал (каналы), при условии, что он обеспечивает взаимоприемлемый уровень секретности. Это позволяет механизмам защиты потоков мультимедийной информации и любым каналам управления функционировать совершенно независимым образом, с совершенно разными уровнями стабильности и сложности.

Если требуется, чтобы канал H.245 функционировал в режиме без шифрования, то конкретные ключи шифрования мультимедийной информации могут шифроваться отдельно по методу, сообщенному участникам и согласованному между ними. Для обеспечения данных для защиты ключей шифрования мультимедийной информации может использоваться логический канал типа **h235Control**. Этот логический канал может работать в любом должным образом согласованном режиме.

Секретность (шифрование) данных, передаваемых в логических каналах, должна соответствовать формату, задаваемому **OpenLogicalChannel**. Данные заголовка для конкретного транспорта не должны зашифровываться. В основе секретности данных должно лежать сквозное шифрование канала.

6.6 Доверительные элементы

Терминалы канала связи определяют основу проведения аутентификации (установления доверительных отношений) и установления секретности. Для канала установления соединения этот процесс может происходить между вызывающим объектом и конечным узлом сети. Например, абонент телефонной связи "верит", что коммутатор сети соединит его с тем номером, который был набран. Поэтому любой объект, находящийся на конце зашифрованного канала управления H.245, или любые логические каналы типа **encryptedData** будут считаться доверительным элементом данного соединения; сюда могут относиться узел(ы) управления многосторонней связью (MCU) и шлюзы. Если элемент признан доверительным, то ему раскрывают механизм секретности (алгоритм и ключ).

Согласно всему вышесказанному, обязанностью лиц, задействованных в передаче по каналу связи, является аутентификация любых и всех "доверительных" элементов. Обычно, для этого производится обмен сертификатами, как и в случае "стандартной" сквозной аутентификации. Настоящая Рекомендация не требует никакого конкретного уровня аутентификации, за исключением, приемлемости ее для всех объектов, использующих этот доверительный элемент. Детали модели доверительности и стратегии сертификатов подлежат дальнейшему изучению.

Обеспечить секретность между двумя конечными точками можно только в том случае, если доказано, что соединения между доверительными элементами действительно защищены от попыток нарушения защиты со стороны посторонних лиц.

6.6.1 Хранение ключей

Хотя специально для функционирования это не требуется, в настоящей Рекомендации содержатся положения для объектов, использующих протокол H.235 для поддержки технической возможности, известной как "доверенная третья сторона" (TTP) в рамках элементов сигнализации.

В тех устройствах, где такая функция желательна или необходима, следует поддерживать возможность восстановления потерянных ключей шифрования мультимедийной информации.

Хранение ключа – это функция, которую часто считают "доверенной третьей стороной" (TTP). Она подлежит дальнейшему изучению.

6.7 Защита от неподтверждения

Требует дальнейшего изучения.

6.8 Защита мобильности

Системы, базирующиеся на H.323 могут применяться в условиях мобильности согласно Рек. МСЭ-Т H.510. Процедуры и протоколы защиты таких систем описаны в Рек. МСЭ-Т H.530. Рек. МСЭ-Т H.530 использует протоколы и процедуры из данной Рекомендации.

6.9 Профили защиты

Эта Рекомендация включает ряд приложений (то есть: Приложения D, E, и F) , каждое из которых содержит профили защиты H.235. Профиль защиты определяет конкретное применение H.235 или подмножества функциональных возможностей H.235 для конкретно определенных условий с широкими возможностями для применения.

В зависимости от условий и возможностей применения профили защиты могут вводиться выборочно или одновременно. Обычно, системы, действующие на основании H.235, указывают в рамках идентификаторов объектов, являющихся составной частью сигнальных сообщений сигнализации, какие профили защиты они используют. Системы H.235 должны выбирать профили защиты согласно своим потребностям.

Дополнительно, конечные точки могут изначально одновременно предложить множество профилей защиты в сообщениях RRQ/GRQ, предоставляя возможность контроллеру доступа выбрать наиболее подходящие из них путем ответа в сообщении RCF/GCF. Транзакции LRQ/LCF между контроллерами доступа также могут переносить некоторые профили защиты. При вычислении цифровых подписей или хеш-значений для обеспечения целостности сообщения сначала необходимо вычислить во всем подмножестве полей хеш-значения и цифровые подписи, которые не обеспечивают целостность сообщения, и поместить их в сообщение, а все цифровые подписи и хеш-значения, которые обеспечивают целостность сообщения, должны быть установлены в 0 в буфере сообщений, затем необходимо вычислить все цифровые подписи и хеш-значения, задействуя этот буфер, и поместить их в сообщение.

7 Процедуры установления соединения

7.1 Введение

Как указывалось в пункте "Реализация системы", как канал установления соединения (H.225.0 для серии H.323), так и канал управления вызовом (H.245) должны работать в согласованном защищенном или незащищенном режиме, начиная с первого обмена. Для канала установления соединения это осуществляется *заранее* [для H.323, для сообщений Q.931 будет использоваться TSAP (порт 1300) с защитой посредством TLS]. Для канала управления вызовом режим защиты определяется посредством информации, передаваемой в протоколе установления первоначального соединения, используемом терминалом серии H

В тех случаях, когда нет возможностей обеспечить "наложенную" защиту, вызываемый терминал может отказать в соединении. Возвращаемое сообщение об ошибке не должно содержать никакой информации относительно несоответствия защиты; вызывающему терминалу придется каким-либо иным образом определять проблему. Если вызывающий терминал получает какое-либо сообщение без надлежащих характеристик защиты, то он должен прекратить вызов.

Если вызывающий и вызываемый терминалы имеют совместимые характеристики защиты, то обе стороны должны будут считать, что канал H.245 должен работать в согласованном режиме защиты. Неудачу при организации работы канала H.245 в определенном здесь режиме защиты следует рассматривать как ошибку протокола, и соединение должно быть разъединено.

8 Сигнализация и процедуры H.245

Обычно аспекты секретности мультимедийных каналов контролируются так же, как и любой другой параметр кодирования; каждый терминал указывает свои характеристики, источник данных выбирает используемый формат, а получатель подтверждает данный режим или отказывается от него. Все независимые от транспорта аспекты такого механизма, например, выбор алгоритма, даются в виде родовых элементов логического канала. Спецификации транспортных средств, такие как – синхронизация с использованием алгоритма/ключа шифрования, передаются в специальных транспортных структурах.

8.1 Функционирование канала H.245 с защитой

Допуская, что процедуры соединения в предыдущем пункте ("Процедуры установления соединения") указывают на защищенный режим работы, то для канала управления H.245 согласованное квитирование и аутентификация должны происходить до того, как состоится обмен какими-либо отличными от H.245 сообщениями. Любой обмен сертификатами, если он согласован, должен будет осуществляться с использованием любого механизма, подходящего для терминала (терминалов) серии H. После завершения установления защиты канала H.245 терминалы используют протокол H.245 так же, как они бы делали это в режиме без защиты.

8.2 Функционирование канала H.245 без защиты

В ином случае канал H.245 может работать в режиме без защиты, при этом два объекта открывают защищенный логический канал, посредством которого выполняется аутентификация и/или извлечение общего "ключа". Например, возможно использование TLS или IPSEC, для чего открывается логический канал с **dataType**, содержащим значение для **h235Control**. С помощью этого канала можно затем получить общий "ключ", который защищает все ключи мультимедийного сеанса, или транспортировать **EncrytionSync**.

8.3 Обмен характеристиками

Согласно процедурам, описанным в 5.2/H.245 ("Процедуры обмена характеристиками"), и соответствующей Рекомендации по системам серии H, конечные точки обмениваются характеристиками с помощью сообщений H.245. Эти наборы характеристик могут в настоящее время включать в себя определения, указывающие параметры шифрования и защиты. Например, конечная точка может сообщать характеристики передачи и приема видеосигналов H.261. Она также может сигнализировать о возможности передачи и приема зашифрованного видеосигнала H.261.

Каждый алгоритм шифрования, который используется вместе с конкретным мультимедийным кодеком, подразумевает определение новой характеристики. Как и для всех остальных характеристик, в процессе своего обмена конечные точки могут сообщать как о независимых, так и о зависимых зашифрованных кодеках. Это позволит конечным точкам изменять свои защитные характеристики с учетом имеющихся ресурсов и дополнительных служебных данных.

По завершении обмена характеристиками конечные точки могут открывать защищенные логические каналы для мультимедийной информации таким же образом, как они бы это сделали при незащищенном режиме.

8.4 Роль ведущего объекта

Схема "ведущий–ведомый" H.245 применяется для введения роли ведущего объекта в целях двунаправленной работы канала и разрешения иных конфликтов. Роль ведущего также используется в методах защиты. Хотя режим (режимы) защиты потока мультимедийной информации устанавливается источником (в отличие от характеристик получателя), ведущей является та конечная точка, которая формирует ключ шифрования. Такое создание ключа шифрования осуществляется независимо от того, является ли ведущий получателем или источником зашифрованной мультимедийной информации. Для того, чтобы обеспечить многопунктовую работу канала многоадресной передачи с применением общих "ключей", MCU (также ведущий) должен формировать такие ключи.

8.5 Передача сигналов по логическому каналу

Конечные точки открывают защищенные мультимедийные логические каналы так же, как они открывают незащищенные мультимедийные логические каналы. Каждый канал может работать совершенно независимо от других – в частности, это относится к защите. Конкретный режим должен определяться в

поле **OpenLogicalChannel dataType**. Исходный ключ шифрования должен передаваться или в **OpenLogicalChannel**, или в **OpenLogicalChannelAck**, в зависимости от соотношения "ведущий–ведомый" отправителя **OpenLogicalChannel**.

OpenLogicalChannelAck должен будет служить подтверждением режима шифрования. Если **OpenLogicalChannel** является неприемлемым для получателя, то в поле причины **OpenLogicalChannelReject** (т. е. причины отказа) должен быть возвращен или **dataTypeNotSupported**, или **dataTypeNotAvailable** (условие перехода).

Во время обмена протоколами, устанавливающего логический канал, ключ шифрования должен передаваться от ведущего терминала к ведомому (независимо от того, кто инициировал **OpenLogicalChannel**). Для мультимедийных каналов, открытых другой конечной точкой (не являющейся ведущей), ведущий терминал должен будет возвратить исходный ключ шифрования и исходную точку синхронизации в **OpenLogicalChannelAck** (в поле **encryptionSync**). Для мультимедийных каналов, открытых ведущим терминалом, **OpenLogicalChannel** должно включать исходный ключ шифрования и точку синхронизации в поле **encryptionSync**.

8.6 Защита быстрого соединения

Конечные точки могут применять процедуру быстрого соединения (см. 8.1.7 и 8.1.7.1/ H.323), используя элемент быстрого старта для защищенного обмена данными ключей (задающего ключа и сеансовых ключей шифрования). Процедуры, представленные в 8.6.1, описывают "ровный" быстрый старт, при котором не используется множество предлагаемых алгоритмов шифрования, тогда как в 8.6.1.1 описывается конкретный пример быстрого старта с использованием множества предлагаемых алгоритмов шифрования, что дает возможность более компактного кодирования сообщений.

8.6.1 Защита однонаправленного быстрого старта

Эта процедура описывает способ создания (полудуплексного) однонаправленного защищенного логического канала от вызывающего объекта к вызываемому.

Процедуры, выполняемые вызывающим объектом

Вызывающий объект (источник сообщения **Setup**) предъявляет как маркер Диффи-Хеллмана (DH), так и обеспечиваемые им структуры FastStart. Маркер DH должен передаваться в рамках вложенного ClearToken в составе CryptoToken, или же, как отдельное ClearToken, см. также 8.8. Во время передачи последовательности **Setup**-to-**Connect** должен осуществляться обмен маркерами Диффи-Хеллмана (DH): это обеспечивает обе конечные точки общим "ключом". Поле **ClearToken** полей **CryptoToken** должно включать **dhkey**, используемый для передачи параметров, что описывается в этой Рекомендации. **halfkey** содержит произвольный открытый ключ одного участника конференции, **modsize** содержит исходный ключ DH, а **generator** содержит DH-группу. Параметры DH, требующие применения, указаны в таблице D.4. За дальнейшими подробностями, пожалуйста, обращайтесь к [Стандарт 2412, Приложение E2].

ПРИМЕЧАНИЕ. – Как только сообщения H.225.0 аутентифицированы (как описано ранее в процедуре I), обмен DH также аутентифицируется.

В любом направлении передачи сообщений о посылке вызова H.225.0, переносящих полуключ Диффи-Хеллмана, когда имеются данные об идентификации, вызывающий или вызываемый объект, если они зарегистрированы, должны также включать отдельный сквозной **ClearToken** с набором **sendersID** в идентификатор конечной точки отправителя и набор **tokenOID**, установленный в "E". Любой промежуточный объект сигнализации H.323 должен передавать этот конкретный сквозной маркер в неизменном виде.

Структуры FastStart блокируют предоставляемые открытые логические каналы с предполагаемыми характеристиками защиты. Должны предоставляться оба канала – H235Cap и nonH235Cap. Во время обмена характеристиками H.245Cap конечные точки предоставляют входные данные **H235SecurityCapability** для обеспечиваемых ими кодеков. Каждому кодеку соответствует отдельная характеристика защиты H.235. Согласно Приложению D, эти характеристики должны указывать на поддержку 128-битового AES-CBC (OID – "Z3"), 56-битового CBC, совместимого с RC2 (OID – "X"), также они должны указывать на поддержку 56-битового DES-CBC (OID – "Y") и могут указывать на поддержку 168-битового тройного DES-CBC (OID – "Z") или 168-битового тройного DES-EOFB (OID – "Z1", совместимого с RC2 EOFB (OID – "X1"), DES-EOFB (OID – "Y1") или AES-EOFB (OID – "Z2"), см. также таблицу D.6.

OpenLogicalChannel передает как **forwardLogicalChannelParameters**, так и **reverseLogicalChannelParameters** с **dataType**, обеспечивая элемент **h235Media** с **encryptionAuthenticationAndIntegrity**, при этом в **encryptionCapability** должен присутствовать, по крайней мере, один **MediaEncryptionAlgorithm**.

В целях взаимосвязи для защиты вызывающий объект априорно является ведущим, см. также 8.4.

Вызывающий объект должен установить истинное значение **mediaWaitForConnect**, чтобы убедиться в том, что данные сеансового ключа доступны, а полученная зашифрованная мультимедийная информация может быть дешифрована. В сценариях, когда необходимо получение "досрочной" мультимедийной информации и вызываемый объект одновременно передает зашифрованную и незашифрованную мультимедийную информацию, посылая при этом сообщение ответа и данные ключа шифрования, вызывающий объект должен быть готов к тому, что он не сможет дешифровать содержимое, пока не получит доступ к данным ключа.

ПРИМЕЧАНИЕ. – В случае, если вызываемый объект посылает зашифрованную мультимедийную информацию вызывающему объекту (что теоретически он может сделать, имея адреса RTP/RTCP вызывающего объекта), вызывающий объект не сможет расшифровать ее не имея общего "ключа", предоставляемого в сообщении Connect (Alerting, Call Proceeding).

Процедуры, выполняемые вызываемым объектом

Во время FastStart вызываемый объект представляет свой маркер DH (см. также 8.8) и приемлемые структуры FastStart. В случае применения процедуры Диффи-Хеллмана рекомендуется, чтобы вызываемый объект при первой же возможности возвращал свой маркер DH в составе сообщения ответа; т. е., сообщения ответа, следующего сразу же за SETUP. Это позволит вызывающему объекту вычислить основной ключ, исходя из общего "ключа" DH, и подготовиться к получению сеансового ключа и зашифрованной мультимедийной информации.

ПРИМЕЧАНИЕ 1. – В случае недоступности алгоритма шифрования на обеих сторонах, поток мультимедийной информации может остаться незашифрованным или же соединение может быть прервано, в зависимости от стратегии защиты.

Каждый объект должен выделять соответствующие наименее значащие биты из общего "ключа" Диффи-Хеллмана для основного ключа шифрования (master key); т. е., 56 наименее значащих битов "ключа" Диффи-Хеллмана для OID "X", OID "X1", OID "Y1" или OID "Y" и 168 наименее значащих битов "ключа" Диффи-Хеллмана для OID "Z", OID "Z1" или OID "Z2", а также 128 наименее значащих битов "ключа" Диффи-Хеллмана для OID "Z3" или OID "Z2", см. также таблицу D.6.

Ответы **OpenLogicalChannel(Ack)** выдаются с помощью сформированного (основного) сеансового ключа, включенного в поле **encryptionSync**. Это **encryptionSync** сохраняет сеансовый ключ для прямого логического канала от вызывающего объекта к вызываемому. Транспортировка ключа должна происходить согласно процедуре, описанной в B.2.4, с использованием или **KeySyncMaterial**, или **V3KeySyncMaterial** (см. B.2.4.1). Сеансовый ключ должен шифроваться с помощью общего "ключа" DH описанным далее способом.

ПРИМЕЧАНИЕ 2. – Не существует никакого предписанного метода для формирования сеансовых ключей, которые применяются для шифрования мультимедийной информации. Формирование этих значений ключей зависит от реализации, на которую влияют местные ресурсы, стратегия и используемый алгоритм шифрования. Нужно соблюдать осторожность, чтобы избежать формирования "нестойких" ключей.

С использованием процедуры из B.2.4, сеансовый ключ шифрования должен быть перенесен в **H.235Key/sharedSecret** в рамках поля **encryptionSync**. Сеансовый ключ должен быть перенесен в поле **keyMaterial** структуры **KeySyncMaterial** и, при условии отсутствия большого числа большеформатных блоков, должен быть добавлен в массив блоков до шифрования. Величина этого дополнения должна определяться посредством обычного согласования алгоритма шифрования. Дополнительная структура **KeySyncMaterial** должна шифроваться с использованием:

– 56 битов общего "ключа", начиная с наименее значащих битов из "ключа" Диффи-Хеллмана при OID "X", OID "X1" , OID "Y1" или OID "Y".

– Все биты общего "ключа" при OID "Z2", OID "Z" или OID "Z1", начиная с наименее значащих битов из "ключа" DH.

Или же, что более предпочтительно, необходимо использовать, если это возможно благодаря процедуре индикации, версии 3 (см. B.2.3), транспортировку уточненного ключа.

В случае, если необходимо организовать полнодуплексный защищенный мультимедийный канал из двух однонаправленных каналов, используя быстрый старт, вызываемый объект должен открыть второй логический канал в направлении вызывающего объекта. Этот логический канал должен передавать

сигналы в отдельном элементе fastStart. Используя имеющийся общий "ключ" DH в качестве основного ключа, вызывающий объект включает в **encryptionSync** другой сеансовый ключ для этого логического канала.

8.6.1.1 Использование множества алгоритмов шифрования при быстром соединении

Согласование алгоритмов шифрования мультимедийной информации в рамках процедур быстрого соединения ведет к необоснованному увеличению количества элементов **OpenLogicalChannel** в элементе **fastConnect** сообщения SETUP. Это происходит вследствие того, что для каждой комбинации кодека (**dataType**) и алгоритма шифрования (включая "none") требуется отдельный **OLC**.

Применяемый для шифрования потока мультимедийной информации алгоритм шифрования устанавливается путем включения в **OLC** структуры **dataType.h235Media.encryptionAuthenticationAndIntegrity.encryptionCapability dataType**. Рек. H.235, версия 2, практически включает только один **MediaEncryptionAlgorithm** в **encryptionCapability**, хотя этот последний элемент определяется как последовательность предшествующих элементов. Эта процедура позволяет включать упорядоченную по приоритетам последовательность характеристик шифрования в каждом предлагаемом **OLC**. Получатель **OLC** должен затем выбрать один алгоритм из предлагаемых и должен возвратить **OLC** с единственным выбранным алгоритмом (вместе с соответствующими транспортными адресами и данными ключей шифрования).

Для того, чтобы добиться максимальной эффективности, ID объекта "NULL-ENCR" (см. таблицу 1) предоставляет алгоритм шифрования "null", что означает полное отсутствие необходимости в операции шифрования. Применение этого конкретного метода требует наличия только одного **OLC** в расчете на каждый предлагаемый кодек и на каждое направление.

Процедура для вызывающего объекта (см. 8.1.7.1/H.323)

Если предлагаемый элемент **dataType** устанавливает режим шифрования посредством выбора **h235Media**, то включенный элемент **encryptionAuthenticationAndIntegrity** может заключать элемент **encryptionCapability**, содержащий множество алгоритмов шифрования (включая алгоритм NULL). Эта конструкция должна быть реализована для того, чтобы дать возможность выбора любого из определенных алгоритмов для шифрования соответствующей мультимедийной информации.

Процедура для вызываемого объекта (см. 8.1.7.1/H.323)

Если множество алгоритмов шифрования предлагается для канала, то вызываемая конечная точка должна выбрать один из них и изменить **OpenLogicalChannel**, чтобы удалить остальные.

Таблица 1/H.235 – Идентификатор объекта при NULL шифровании

Эталонное значение идентификатора объекта	Значение идентификатора объекта	Описание
"NULL-ENCR"	{itu-t (0) recommendation (0) h (8) 235 version (0) 3 26}	Указывает на "Алгоритм шифрования NULL."

8.6.2 Защита двунаправленного быстрого старта

Вопрос защиты двунаправленных каналов передачи данных T.120 подлежит дальнейшему изучению.

8.7 Зашифрованные DTMF сигналы H.245

Конечные точки для достижения конфиденциальности могут отдать предпочтение отправке шифрованных DTMF сигналов. Используя сеансовые ключи шифрования, конечные точки могут зашифровывать DTMF сигналы в **UserInputIndication** следующим образом:

- Зашифрованная основная строка: **encryptedAlphanumeric**;
- Зашифрованная строка iA5: **encryptedSignalType** в рамках **signal**;
- Зашифрованная общая строка: **encryptedAlphanumeric** в рамках **extendedAlphanumeric**.

ПРИМЕЧАНИЕ 1. – Дополнительные параметры для RTP в строке iA5 с временными метками и номерами логических каналов или обновленные сигналы с длительностью тональных сигналов не шифруются, так как считается, что они не передают критическую информацию.

Согласованная характеристика **secureDTMF** относится к шифрованной строке iA5.

Управление ключами, описанное в Приложении D.7, должно применяться для формирования сеансового ключа шифрования. Этот сеансовый ключ шифрования должен использоваться для шифрования H.245 DTMF сигналов.

ПРИМЕЧАНИЕ 2. – Это не обязательно предполагает применение этого сеансового ключа и для шифрования полезной нагрузки RTP.

Тем не менее, при использовании также и DTMF сигналов посредством RTP и путем установки флага **rtpPayloadIndication** настоятельно рекомендуется защищать полезную нагрузку RTP, используя профиль шифрования речевых сообщений из Приложения D.7.

В таблице 2 представлены имеющиеся алгоритмы шифрования (DES, 3DES или AES), которые должны использовать EOFB (включая OFB в качестве особого случая, см. B.2.5). Для того, чтобы избежать потенциальной вставки символов DTMF, не рекомендуется для шифрования DTMF применять CBC, CFB или другие режимы формирования последовательности блоков передаваемой информации, что может потребовать вставки.

8.7.1 Зашифрованная основная строка

Если выбран элемент **encryptedBasicString** в **UserInputCapability**, то тогда **encryptedAlphanumeric** должно будет указывать на применяемый алгоритм шифрования в рамках **algorithmOID**, а **paramS –** содержать исходное значение для операции шифрования. Зашифрованная буквенно-цифровая строка должна быть помещена в **encrypted**.

8.7.2 Зашифрованная строка iA5

Если выбран элемент **encryptedIA5String** в **UserInputCapability**, то тогда **encryptedSignalType** должен будет содержать шифрованный **ClearSignalType**, где **sig** переносит символ **signalType** нешифрованного текста. Элемент **signalType** должен будет содержать фиктивный "!", который должен быть аннулирован получателем.

Элемент **algorithmOID** должен указывать на применяемый алгоритм шифрования, **paramS** – содержать исходное значение для операции шифрования.

8.7.3 Зашифрованная общая строка

Если выбран элемент **encryptedGeneralString** в **UserInputCapability**, то тогда элемент **encryptedAlphanumeric** в рамках **extendedAlphanumeric** должен указывать на применяемый алгоритм шифрования в рамках **algorithmOID**, тогда как **alphanumeric** должен содержать пустую строку, а **paramS** – содержать исходное значение для операции шифрования.

8.7.4 Перечень идентификаторов объектов

Таблица 2/H.235 – Идентификаторы объектов для шифрования DTMF сигналов H.245

Эталонное значение идентификатора объекта	Значение идентификатора объекта	Описание
"DES-EOFB-DTMF"	{itu-t (0) recommendation (0) h (8) 235 version (0) 3 12}	Шифрование DTMF сигнала H.245 посредством DES-56 в режиме EOFB
"3DES-EOFB-DTMF"	{itu-t (0) recommendation (0) h (8) 235 version (0) 3 13}	Шифрование DTMF сигнала H.245 посредством 3DES-168 в режиме EOFB
"AES-EOFB-DTMF"	{itu-t (0) recommendation (0) h (8) 235 version (0) 3 14}	Шифрование DTMF сигнала H.245 посредством AES-128 в режиме EOFB

8.8 Функционирование системы Диффи-Хеллмана

В данной Рекомендации для сквозного согласования ключей поддерживается протокол Диффи-Хеллмана. В зависимости от ситуации, согласованный ключ Диффи-Хеллмана может действовать как основной ключ (Приложение D.7) или же, как динамический сеансовый ключ (Приложение F и Рек. МСЭ-Т H.530).

Система Диффи-Хеллмана характеризуется системными параметрами g и p, где p – универсальный основной ключ, а g обозначает генератор групп мультипликативных сигналов по модулю p или подгрупп сигналов с высоким уровнем по модулю p. g^x по модулю p обозначает (открытый) полуключ Диффи-Хеллмана вызывающего объекта, тогда как g^y по модулю p обозначает (открытый) полуключ Диффи-Хеллмана вызываемого объекта. Стандарт RFC 2412 дает дополнительную исходную информацию и консультации в части того, как выбрать параметры защиты Диффи-Хеллмана.

В Рек. МСЭ-Т H.235 представлен экземпляр системы Диффи-Хеллмана с параметрами (g, p, g^x), кодированными в рамках **ClearToken**, причем **dhkey** содержит **halfkey** g^x по модулю p (соответ. g^y по модулю p) для некоего секретного произвольного x (соответ. y), основной p в **modsize** и **generator** g. Особый случай – это тройной (0, 0, 0) или пустой **dhkey**, который не представляет экземпляра DH, но который должен использоваться при сигнализации о том, что профиль шифрования речевых сообщений не используется.

Часто параметры p и g системы DH являются постоянными для ряда приложений и имеют четкие значения, хотя конечные системы могут также выбрать свой собственный набор параметров. Вызываемый объект должен быть осведомлен о том, что нестандартные параметры DH могут в меньшей степени обеспечить защиту, чем те параметры, которые на первый взгляд выглядят как правомочные; к примеру, вызывающий объект может выбрать не основной ключ, или же g генерирует меньшую подгруппу. До тех пор, пока комплексное тестирование параметров практически недостижимо, вопрос о том, принимать или отвергать такие предложения, относится к области стратегии защиты.

При фиксированных параметрах системы DH поверхностное определение параметров посредством идентификатора объектов может потребовать более компактных кодированных сообщений, чем те, что включают буквальные значения. Элемент **ClearToken**, который переносит экземпляр DH с фиксированными, стандартизированными параметрами DH, может указывать на этот экземпляр DH-OID в поле **tokenOID**, если только **tokenOID** не используется для других целей (таких, как описываются в D.6.3.2 для выделенного элемента **CryptoToken**). Отправитель может дополнительно включать буквальные значения DH, но в этом нет необходимости.

В случае, если должны быть указаны несколько экземпляров DH, причем, каждый посредством DH-OID, параметры DH в выделенном **CryptoToken** (которому посвящено Приложение D) должны быть опущены путем дальнейшего отсутствия **dhkey**, а все экземпляры DH должны затем переноситься в рамках отдельных **ClearTokens**, при этом элемент **tokenOID** содержит DH-OID, а **dhkey** может и далее отсутствовать; все остальные поля в рамках этого элемента **ClearToken** не должны использоваться.

ПРИМЕЧАНИЕ 1. – Это не предусматривает возможности передачи экземпляра DH в выделенном **CryptoToken** или других имеющихся **ClearTokens** путем явного включения значений параметров DH.

В случае, если нужно указать на нестандартный экземпляр DH, надо использовать DH-OID "DHdummy", а нестандартные параметры группы DH должны быть четко представлены в **ClearToken**.

Вызывающий объект может представить на рассмотрение один или несколько **ClearTokens**, причем каждый переносит другой экземпляр Диффи-Хеллмана. Вызывающий объект может представлять столько экземпляров DH, сколько позволяет его стратегия защиты. Это позволяет вызываемому объекту выбрать соответствующий экземпляр для ответа, увеличивая, таким образом, вероятность нахождения правильного общего набора параметров.

Вызываемый объект должен выбрать и принять один экземпляр DH (если это вообще произойдет), который он выбирает из неупорядоченного набора экземпляров DH, предоставляемых вызывающим объектом в сообщении SETUP. В случае если вызываемый объект способен выбрать такой экземпляр DH, который соответствует его потребностям в защите, этот вызываемый объект не должен изменять предлагаемый экземпляр DH или возвращать тот, что был послан вызывающим объектом. Устойчивость алгоритмов шифрования, доступных каждой конечной точке (EP) во время соединения, должна соответствовать устойчивости, обеспечиваемой выбранным экземпляром DH, который возвращается вызываемым объектом; см. таблицу D.4. Вызываемый объект должен указывать выбранный экземпляр DH в сообщении ответа.

В случае, если вызываемый объект отвергает любые предложения по причинам безопасности или вследствие недостаточных возможностей обработки вызова, то вызываемый объект должен опустить **dhkey** в сообщении ответа.

Вызываемый объект должен включать свой маркер DH в ответ **Setup**-to-**Connect**. Вызываемый объект может включать свой маркер DH в немедленное сообщение ответа, следующее за SETUP, или же он может включать маркер DH на более позднем этапе, но не позднее сообщения CONNECT.

ПРИМЕЧАНИЕ 2. – Необходимо учитывать некоторые аспекты, касающиеся моментов возможного включения маркера(ов) DH во время ответов **Setup**-to-**Connect**: время ответа, нагрузку вызываемого объекта при обработке вызова, возможность преждевременной передачи мультимедийной информации и другие аспекты. Эти вопросы считаются зависящими от реализации.

По некоторым причинам, однако, некоторые маршрутизирующие GK могут не передавать все ответы **Setup**-to-**Connect** вызывающему объекту. Таким образом, два или несколько ответных сообщений о посылке вызова H.225.0, включая возможный маркер DH, могут быть сброшены и они не поступят к вызывающему объекту. Вызывающий объект не сможет тогда определить основной ключ DH и сеансовый мультимедийный ключ(и). Чтобы предупредить такие случаи, вызываемый объект должен будет всегда включать один и тот же маркер DH в каждое ответное сообщение **Setup**-to-**Connect**.

В случаях, когда DH-OID указывает на другой экземпляр DH, а не на тот, что фактически передается в рамках **modsize** и **generator**, буквальные значения, передаваемые в рамках **modsize** и **generator**, должны будут иметь приоритет над маркером DH-OID. Для ответа вызываемый объект должен заменить конфликтующий DH-OID статическим DH-OID, к примеру, "DH1024," который соответствует **modsize** и **generator** или "DHdummy", если соответствующий DH-OID отсутствует.

9 Процедуры проведения многосторонних конференций

9.1 Аутентификация

Аутентификация должна происходить между конечной точкой и MC(U) таким же образом, как и при двусторонней конференции. MC(U) должен определять стратегию в части уровня и точности аутентификации. Согласно пункту 6.6, MC(U) является доверительным элементом; конечные точки, участвующие в конференции, могут быть ограничены уровнем аутентификации, применяемым MC(U). Новые команды **ConferenceRequest**/**ConferenceResponse** позволяют конечным точкам получить сертификаты других участников конференции от MC(U). Как указано в процедурах H.245, конечные точки в многосторонней конференции могут через MC запросить сертификаты других конечных точек, но, возможно, они не смогут выполнить прямую криптографическую аутентификацию в рамках канала H.245.

9.2 Секретность

MC(U) должен играть главную роль во всех обменах ведущий/ведомый и, при этом, должен обеспечивать ключ(и) шифрования для участников многосторонней конференции. Секретность для отдельных источников в рамках общего сеанса (предположительно многоточечного) может достигаться посредством индивидуальных или общих ключей. Эти два режима могут быть произвольно выбраны MC(U), и ими нельзя будет управлять с какой-либо конкретной конечной точки, это не касается режимов, разрешенных согласно стратегии MC(U). Иными словами, общий ключ может использоваться для множества логических каналов, открытых из различных источников.

10 Передача сигналов аутентификации и процедуры аутентификации

10.1 Введение

Аутентификация обычно базируется или на использовании общего "ключа" (вас правильно идентифицируют, если вы знаете "ключ"), или на применении методов открытого ключа с сертификацией (вы доказываете свою тождественность тем, что владеете правильным личным ключом). Общий "ключ" и последующее использование симметричной криптографии требуют предварительного контакта между объектами, участвующими в процессе связи. Вместо предварительного непосредственного или косвенного контакта можно сформировать общий "ключ" или обмениваться им с помощью методов, базирующихся на криптографии с открытым ключом, например, посредством обмена ключами Диффи-Хеллмана. Необходимо проводить аутентификацию участников процесса связи при формировании ключей и обмене, например, посредством сообщений с цифровыми подписями; в противном случае участники процесса связи не могут быть уверены в том объекте, с которым они имеют общий "ключ".

В настоящей Рекомендации описаны методы аутентификации, базирующиеся на подписке (т. е. для создания общего "ключа" необходим предварительный контакт), и методы аутентификации, при

которых криптография с открытым ключом применяется или непосредственно при аутентификации, или для формирования общего "ключа".

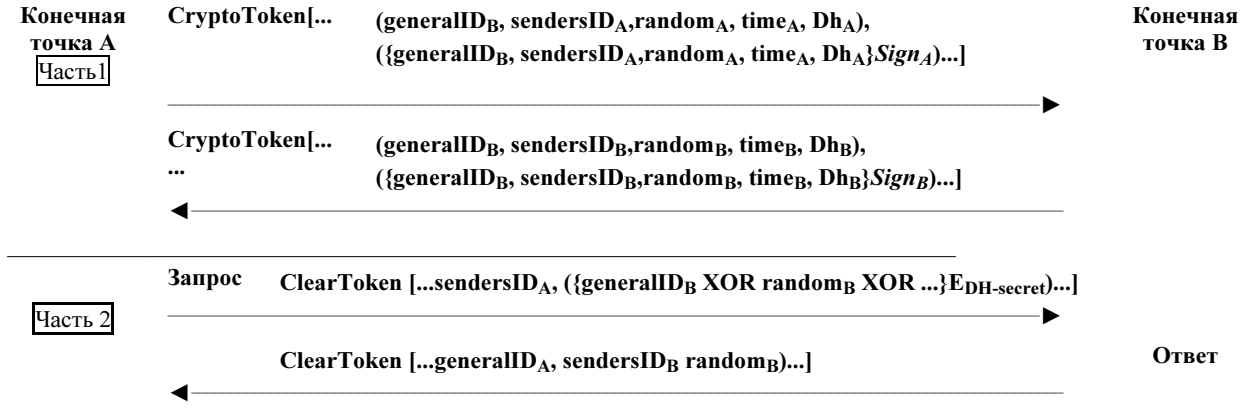
10.2 Применение ключа Диффи-Хеллмана при необязательной аутентификации

Здесь не ставится задача обеспечения абсолютной аутентификации на уровне пользователя. Данный метод обеспечивает передачу сигналов для формирования общего "ключа", который известен двум объектам, и может привести к манипуляциям с данными ключа в случае конфиденциальной связи.

В конце этого обмена оба эти объекта будут владеть общим секретным "ключом", а также выбранным алгоритмом использования этого ключа. Этот общий секретный "ключ" можно теперь использовать при любых дальнейших обменах запрос/ответ. Следует отметить, что в редких случаях для определенных алгоритмов обмен ключами Диффи-Хеллмана может привести к формированию ключей, известных как *неустойчивые* ключи. Когда это происходит, любой из объектов должен отключаться и снова подключиться, чтобы сформировать новый набор ключей.

В первой части рисунка 1 показаны данные, обмен которыми происходил при применении ключей Диффи-Хеллмана. Во второй части приведены сообщения запроса конкретного приложения или протокола, которые должны быть аутентифицированы респондентом. Отметим, что с каждым ответом может возвращаться новая случайная величина.

ПРИМЕЧАНИЕ. – Если происходит обмен сообщениями по незащищенному каналу, то необходимо использовать цифровые подписи (или иной метод аутентификации источника сообщения) для того, чтобы аутентифицировать участников конференции, которые будут иметь общий "ключ". Также может быть предусмотрен необязательный элемент подписи; что показано ***жирным курсивом*** ниже.



[... ...] – последовательность маркеров.

() – конкретный маркер, который может содержать множество элементов.

{} – $\mathbf{E_{DH\text{-}secret}}$ показывает, что содержащиеся там значения зашифрованы с применением ключа Диффи-Хеллмана.

Конечная точка B знает, какой общий "ключ" использовать, чтобы расшифровать идентификатор $\mathbf{generalID_B}$, сопоставив его с $\mathbf{generalID_A}$, который также должен передаваться в сообщении как $\mathbf{sendersID_A}$. Отметим, что зашифрованное значение в части 2 передается в поле **generalID** элемента **clearToken** для упрощения шифрования.

Рисунок 1/H.235 – Применение ключа Диффи-Хеллмана при необязательной аутентификации

10.3 Аутентификация на основе подписки

10.3.1 Введение

Хотя по своей сущности, изложенные здесь процедуры (и алгоритмы ISO, исходя из которых они получены) являются двунаправленными, они могут применяться только в одном направлении, если аутентификация необходима только в этом направлении. Описаны как двухпроходная, так и трехпроходная процедуры. Взаимная двухпроходная аутентификация может осуществляться только в одном направлении, когда не требуется идентифицировать сообщения, идущие в обратном направлении. При таком обмене предполагается, что каждая конечная точка владеет некоторым хорошо известным идентификатором (например, текстовым идентификатором), который обеспечивает ее однозначную идентификацию. Для двухпроходной процедуры делается дополнительное предположение относительно

взаимоприемлемой привязки ко времени (по которой определяются временные метки). Величина приемлемого сдвига по времени – это вопрос конкретной реализации. Трехпроходная процедура использует генерируемый по случайному закону непредсказуемый номер вызова (который может быть дополнен последовательным счетчиком 'random') в виде вызова от идентифицирующей стороны. Этот случайный номер предназначен для защиты от повторных попыток нарушения защиты. В отличие от двухпроходных процедур, трехпроходные процедуры не производят идентификацию первого исходного сообщения, содержащего вызов инициирующей стороны.

В зависимости от требований возможна реализация трех различных вариантов:

1) вариант, основанный на пароле с симметричным шифрованием;

2) вариант, основанный на пароле с хешированием;

3) вариант, основанный на сертификате с подписями.

Во всех случаях маркер будет содержать информацию, соответствующую описанию в следующих пунктах и зависящую от выбранного варианта. Отметим, что во всех случаях **generalID** может быть скорее известен исходя из поиска по каталогу или конфигурации, а не исходя из внутриполосного обмена протоколами. Для упрощения обработки на стороне получателя отправитель должен включить свой идентификатор в **sendersID** и установить **generalID** в соответствии с идентификацией получателя.

ПРИМЕЧАНИЕ 1. – Во всех случаях, когда временные метки формируются и передаются как составляющая конфиденциального обмена, при реализации следует принимать следующие меры предосторожности. Детализация временной метки должна быть достаточно мелкой, чтобы с каждым сообщением она гарантированно возрастала. Если это не гарантируется, то возможны повторные попытки нарушения защиты (например, если временная метка увеличивается только поминутно, то конечная точка "C" может обманывать конечную точку "A" в течение одной минуты с того момента, как конечная точка "A" отправила сообщение конечной точке "B").

ПРИМЕЧАНИЕ 2. – Если сообщение многоадресное, то оно не защищено.

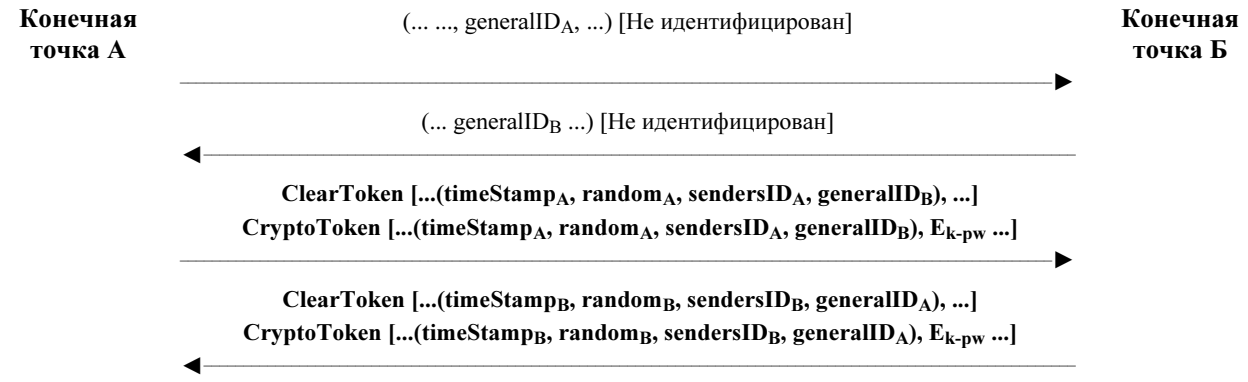
10.3.2 Пароль с симметричным шифрованием

На рисунках 2a и 2b показан формат маркера и обмен сообщениями, необходимые для выполнения аутентификации такого типа, соответственно, двумя или тремя проходами. Этот протокол основывается на пунктах 5.2.1 (двухпроходная процедура) и 5.2.2 (трехпроходная процедура) Рекомендации ИСО/МЭК 9798-2; предполагается, что при подписке производится обмен идентификатором и соответствующим паролем. Длина ключа шифрования составляет N октетов (как указано в AlgorithmID), сформирован этот ключ следующим образом:

– если длина пароля = N, то ключ = пароль;

– если длина пароля < N, то в ключ добавляются нули;

– если длина пароля > N, то первые N октетов присваиваются этому ключу, затем производится логическая операция "исключающее ИЛИ" над N + M-*ным* октетом и Mmod(N)-*ным* октетом (для всех октетов больше N) (т. е. все "излишние" октеты пароля повторно изменяются обратно по ключу с помощью операции "исключающее ИЛИ").



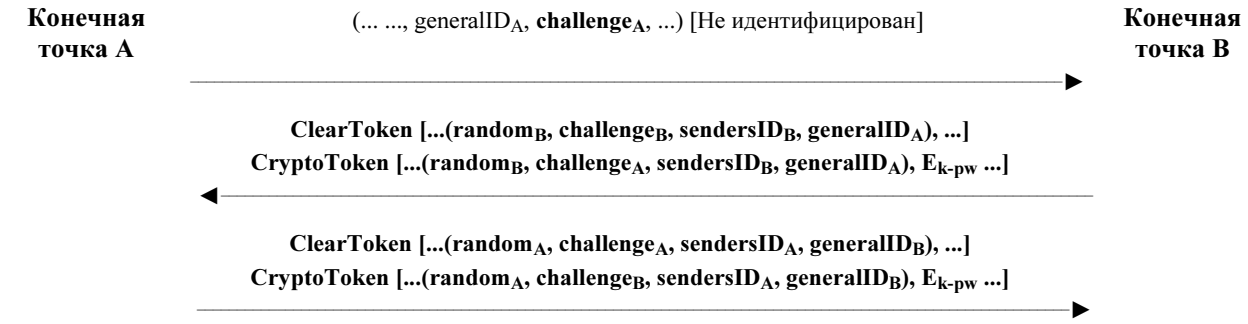
ПРИМЕЧАНИЕ 1. – Маркер, возвращаемый из конечной точки B, является необязательным; если он пропущен, то осуществляется только односторонняя аутентификация.

ПРИМЕЧАНИЕ 2. – **E_{k-pw}** показывает значения, которые зашифрованы посредством ключа "k", полученного из пароля "pw".

ПРИМЕЧАНИЕ 3. – **random** – это равномерно увеличивающийся счетчик, благодаря которому многократно повторяющееся сообщение с одной и той же временной меткой становится уникальным.

ПРИМЕЧАНИЕ 4. – В третьем сообщении конечная точка А создает отдельный маркер **ClearToken**, который идентифицируется посредством того же идентификатора объекта OID, что и OID в **CryptoToken**; то же самое относится к четвертому сообщению и наоборот.

Рисунок 2a/H.235 – Пароль с симметричным шифрованием; два прохода



ПРИМЕЧАНИЕ 1. – **challenge$_A$** и возвращаемый зашифрованный **CryptoToken** от B к A не обязательны, если требуется односторонняя аутентификация.

ПРИМЕЧАНИЕ 2. – **E_{k-pw}** показывает функцию шифрования, которая зашифрована посредством ключа "k", полученного из пароля "pw".

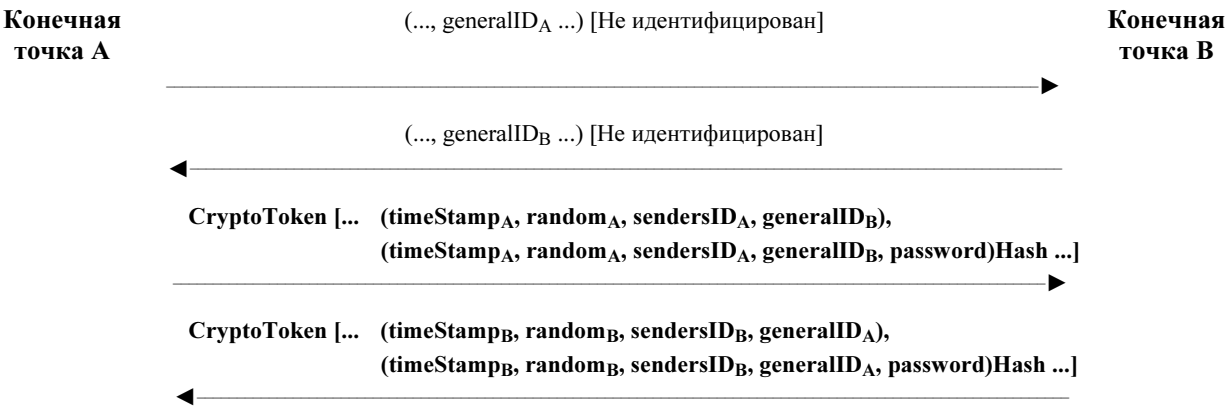
ПРИМЕЧАНИЕ 3. – В третьем сообщении конечная точка А создает новый **challenge$_A$** в незашифрованном тексте в отдельном **ClearToken**, который идентифицируется посредством того же OID, что и OID в **CryptoToken**. Конечная точка А также возвращает зашифрованный **challenge$_B$** в качестве ответа; то же самое относится ко второму сообщению и наоборот.

ПРИМЕЧАНИЕ 4. – В случае множества ожидающих обработки сообщений, **random** (т. е. равномерно возрастающий счетчик) должен сделать вызов уникальным.

Рисунок 2b/H.235 – Пароль с симметричным шифрованием; три прохода

10.3.3 Пароль с хешированием

На рисунках 3a и 3b показаны формат маркера и обмен сообщениями, необходимые для выполнения аутентификации данного типа, соответственно, для двух проходов или трех проходов. Настоящий протокол основывается на пунктах 5.2.1 и 5.2.2 Рекомендации ИСО/МЭК 9798-4; предполагается, что при подписке производится обмен идентификатором и соответствующим паролем. В Приложении D приведено подробное описание двухпроходной процедуры хеширования.

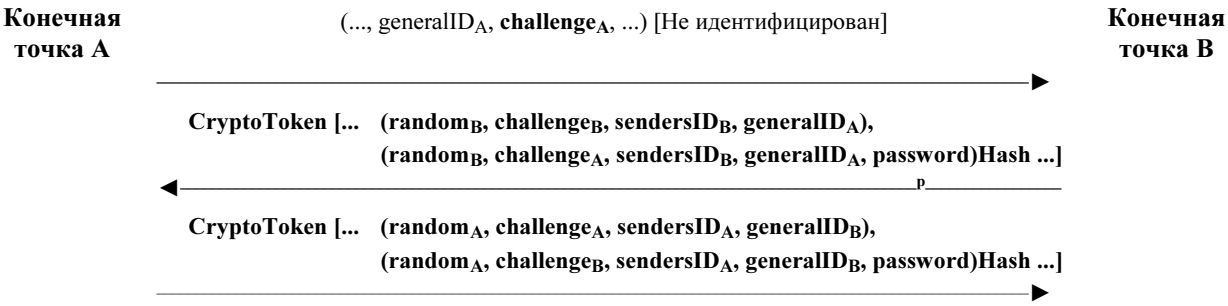


ПРИМЕЧАНИЕ 1. – Маркер, возвращаемый из конечной точки B является необязательным; если он пропущен, то осуществляется только односторонняя аутентификация.

ПРИМЕЧАНИЕ 2. – **Hash** обозначает функцию хеширования, которая оперирует с составляющими ее величинами.

ПРИМЕЧАНИЕ 3. – **random** – это равномерно возрастающий счетчик, благодаря которому многократно повторяющееся сообщение с одной и той же временной меткой становится уникальным.

Рисунок 3a/H.235 – Пароль с хешированием; два прохода



ПРИМЕЧАНИЕ 1. – Маркер, возвращаемый из конечной точки B является необязательным; если он пропущен, то осуществляется только односторонняя аутентификация.

ПРИМЕЧАНИЕ 2. – **Hash** обозначает функцию хеширования, которая оперирует с составляющими ее величинами.

ПРИМЕЧАНИЕ 3. – В третьем сообщении конечная точка A создает новый **challenge$_A$** в незашифрованном тексте в рамках вложенного **ClearToken** в **cryptoHashedToken**. Конечная точка A также возвращает хешированный **challenge$_B$** в качестве ответа; то же самое относится ко второму сообщению и наоборот.

ПРИМЕЧАНИЕ 4. – В случае множества ожидающих обработки сообщений, **random** (т. е. равномерно возрастающий счетчик) должен делать вызов уникальным.

Рисунок 3b/H.235 – Пароль с хешированием; три прохода

ПРИМЕЧАНИЕ 1. – Структура **cryptoHashedToken** применяется для пересылки параметров, используемых в данном обмене. В эту структуру включены 'чистые' версии параметров, необходимых для вычисления хешированного значения. При реализации в **hashedVals** должна быть включена временная метка и *не* должен быть включен пароль. (Например, как пароль, так и '**generalID**' должны быть *заранее* известны получателю; первый из них может быть опущен.)

ПРИМЕЧАНИЕ 2. – Функция хеширования должна будет применяться к структуре **EncodedGeneralToken**, включающей, как минимум, поля идентификатора ID, временной метки и пароля. Значение пароля НЕ должно передаваться в **ClearToken**.

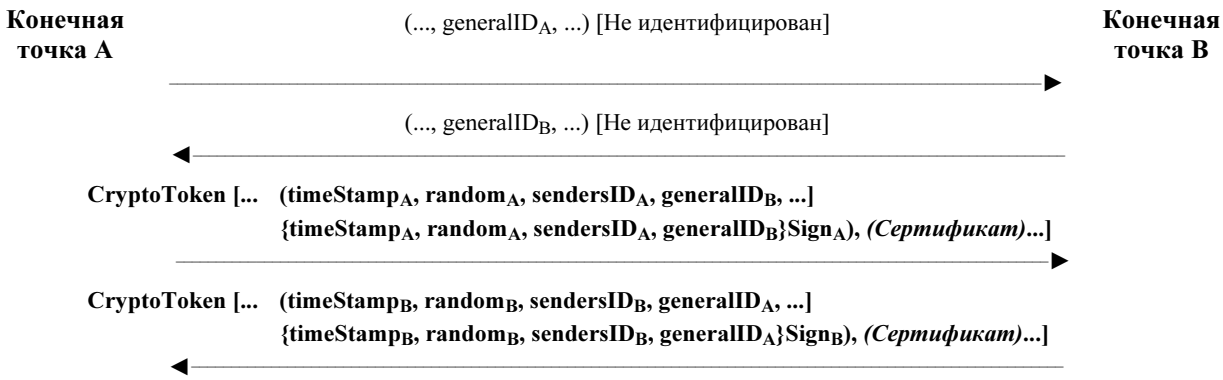
ПРИМЕЧАНИЕ 3. – При реализации этого следует убедиться в том, что пароли, вводимые пользователем, имеют достаточную энтропию. Следует отказываться от слишком коротких паролей или от тех паролей, которые чувствительны к попыткам нарушить защиту структуры данных. В определенных случаях может быть целесообразно, чтобы пользователь вводил фразу – пароль, используя криптографическую функцию хеширования, и использовать выходные биты.

10.3.4 Аутентификация на основе сертификатов с подписями

На рисунках 4a и 4b показаны формат маркера и обмен сообщениями, необходимые для выполнения аутентификации данного типа. Настоящий протокол основывается на пункте 5.2.1 Рекомендации МСО/МЭК 9798-3; предполагается, что при подписке производится присвоение/обмен идентификатором и соответствующим сертификатом. В Приложении E приведено подробное описание двухпроходной процедуры с подписями.

ПРИМЕЧАНИЕ 1. – Может быть предусмотрен необязательный элемент сертификата, что показано ***жирным курсивом*** ниже.

ПРИМЕЧАНИЕ 2. – Если сообщение многоадресное, то идентификатор назначения (**generalID$_B$** для сообщений, исходящих из A, и наоборот) не следует включать в **ClearToken**.



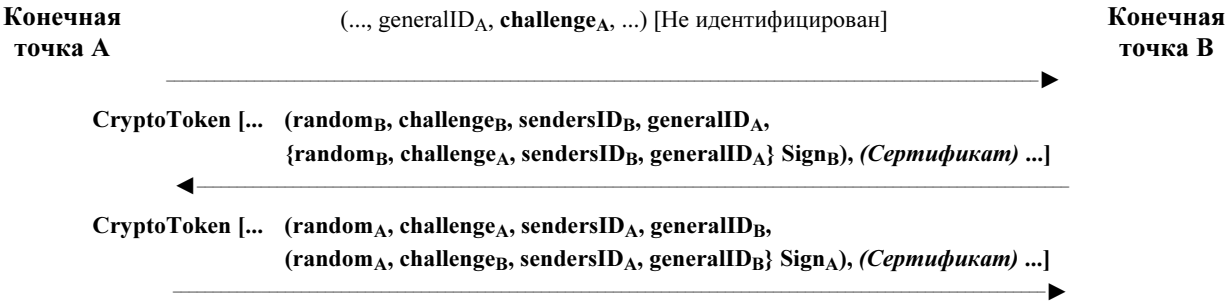
ПРИМЕЧАНИЕ 1. – Маркер, возвращаемый от EPB является необязательным; если он пропущен, то осуществляется только односторонняя аутентификация.

ПРИМЕЧАНИЕ 2. – Сертификат "платного" ("payment") типа может быть дополнительно включен инициатором сообщения в EPA.

ПРИМЕЧАНИЕ 3. – **Sign** означает функцию формирования подписи (из соответствующего сертификата), применяемую к составляющим ее величинам.

ПРИМЕЧАНИЕ 4. – **random** – это равномерно возрастающий счетчик, благодаря которому многократно повторяющееся сообщение с одной и той же временной меткой становится уникальным.

Рисунок 4a/H.235 – Аутентификация на основе сертификатов с подписями; два прохода



ПРИМЕЧАНИЕ 1. – Маркер, возвращаемый из EPB является необязательным; если он пропущен, то осуществляется только односторонняя аутентификация.

ПРИМЕЧАНИЕ 2. – Сертификат "платного" ("payment") типа может быть дополнительно включен инициатором сообщения в EPA.

ПРИМЕЧАНИЕ 3. – **Sign** означает функцию формирования подписи (из соответствующего сертификата), применяемую к составляющим ее величинам.

ПРИМЕЧАНИЕ 4. – В третьем сообщении EPA создает новый **challenge$_A$** в незашифрованном тексте в рамках вложенного зашифрованного **GeneralToken**. EPA также возвращает подписанный **challenge$_B$** в качестве ответа; то же самое относится ко второму сообщению и наоборот.

ПРИМЕЧАНИЕ 5. – В случае множества ожидающих обработки сообщений, **random** (т. е. равномерно возрастающий счетчик) должен делать вызов уникальным.

Рисунок 4b/H.235 – Аутентификация на основе сертификатов с подписями; три прохода

10.3.5 Использование общего "ключа" и паролей

В настоящей Рекомендации определенные симметричные криптографические средства применяются для обеспечения аутентификации, целостности и конфиденциальности. В данном тексте при рассмотрении симметричных методов используются термины "пароль" и общий "ключ" 21. Общий "ключ" понимается как общий термин, обозначающий произвольную цепочку битов. Общий "ключ" может присваиваться или

конфигурироваться как составляющая процесса подписки пользователя, или же он может являться составляющей внутриполосных вычислений, как, например, общий "ключ" Диффи-Хеллмана.

Пароль может рассматриваться как буквенно-цифровая цепочка символов, которую пользователи могут запомнить. Очевидно, что применять пароли следует с осторожностью. Пароли могут обеспечить достаточную защиту только в том случае, если они выбираются произвольно из большого числа паролей, имеют достаточную энтропию, такую, что они являются непредсказуемыми и периодически меняются. Правила формирования, и поддержания паролей выходят за рамки настоящей Рекомендации.

Оптимальный метод, позволяющий использовать преимущества паролей и общих "ключей", заключается в преобразовании цепочки пароля пользователя в фиксированную цепочку битов в виде общего "ключа", используя при этом одностороннюю, криптографически стойкую функцию хеширования.

Как рекомендует следующий пример, при использовании профиля защиты из Приложения D, алгоритм SHA-1 при применении его к цепочке пароля сформирует 20-байтовый общий "ключ". Преимущество состоит в том, что хешированный результат не только скрывает реальный пароль, но также определяет формат цепочки битов фиксированной длины, фактически не жертвуя при этом энтропией.

Таким образом,

Общий "ключ": = SHA1 (пароль)

11 Процедуры шифрования потока мультимедийной информации

Потоки мультимедийной информации должны шифроваться с помощью алгоритма и ключа, как это представлено в канале H.245. На рисунках 5 и 6 показан общий поток. Отметим, что заголовок транспорта присоединяется к транспортному SDU после того, как этот SDU был зашифрован. Непрозрачные сегменты обозначают секретность. Когда передающая сторона получает новые ключи и начинает их использовать при шифровании, заголовок SDU каким-либо образом должен указать получателю, что теперь применяется новый ключ. Например, согласно Рек. МСЭ-Т H.323 заголовок RTP (SDU) изменит свой тип полезной нагрузки, чтобы указать на переход к новому ключу.

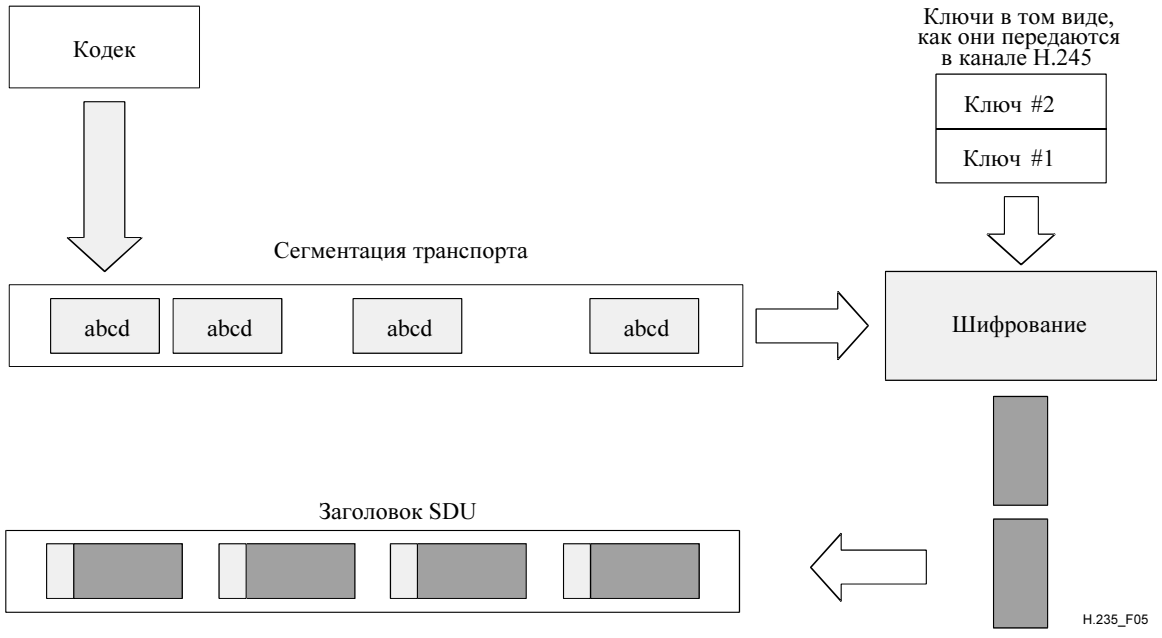


Рисунок 5/H.235 – Шифрование мультимедийной информации

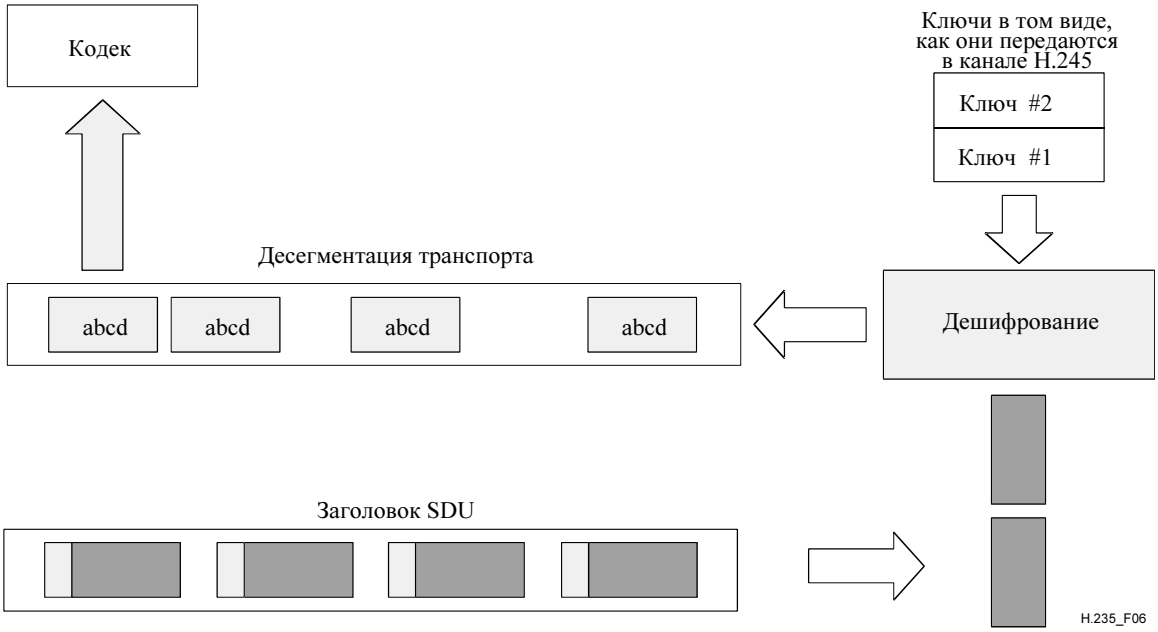


Рисунок 6/H.235 – Дешифрование мультимедийной информации

11.1 Сеансовые ключи для мультимедийной информации

В **encryptionUpdate** включается **h235Key**. **h235Key** кодируется в ASN.1 в контексте дерева ASN.1 H.235 и, в случае H.245, передается как непрозрачная цепочка октетов. Для защиты ключа может использоваться один из трех возможных механизмов согласно передаче сообщений между двумя конечными точками.

- Если канал H.245 защищен, то к данным ключей не применяется никакая дополнительная защита. Для данного поля ключ передается "открытым текстом"; применяется вариант **secureChannel** в ASN.1.
- Если секретный ключ и алгоритм созданы, в целом, за пределами канала H.245 (т. е. вне H.323 или в логическом канале **h235Control**), то для шифрования данных ключа используется общий "ключ"; сюда включается и результирующий зашифрованный ключ. В этом случае применяется вариант **sharedSecret** в ASN.1.
- Когда канал H.245 не является защищенным, могут также применяться сертификаты, но они могут также дополнительно использоваться в случае защищенного канала H.245. При применении сертификатов данные ключа зашифровываются посредством открытого ключа сертификата и конструкции **certProtectedKey** в ASN.1.

В любой точке, участвующей в конференции, получатель (или отправитель) может затребовать новый ключ (**encryptionUpdateRequest**). Одной из причин этого могут послужить возникшие подозрения относительно потери этой стороной синхронизации одного из логических каналов. Ведущий терминал, получивший этот запрос, должен сформировать новый ключ(и) в ответ на эту команду. Ведущий терминал может также решить асинхронно распределять новый ключ(и), в таком случае он должен будет использовать сообщение **encryptionUpdate**.

По получении **encryptionUpdateRequest**, ведущий терминал должен послать **encryptionUpdate**. Если конференция – многосторонняя, то MC (также ведущий) должен распределить новый ключ всем пользователям прежде, чем передать этот ключ отправителю. Отправитель данных по логическому каналу должен использовать новый ключ как можно раньше после получения сообщения.

Отправитель (предполагается, что это не ведущий терминал) может также запросить новый ключ. Если отправитель – участник многосторонней конференции, то будет выполняться следующая процедура:

- Отправитель должен передать **encryptionUpdateRequest** к MC (ведущий терминал).

- МС должен сформировать новый ключ(и) и передать сообщение **encryptionUpdate** всем участникам конференции, за исключением отправителя.
- После распределения новых ключей всем другим участникам МС должен передать **encryptionUpdate** отправителю. Затем отправитель должен использовать новый ключ.

11.2 Защита от спама при передаче мультимедийной информации

Получатель потока мультимедийной информации в RTP может захотеть противостоять попыткам нарушения защиты типа "отказ в обслуживании" и "лавинная передача" в обнаруженных портах RTP/UDP. Получатели, реализовав возможность защиты от спама, могут быстро определить, не исходит ли полученный пакет RTP от неправомочного пользователя, и сбросить его.

Возможность защиты от спама, если она установлена, указывает на использование механизма защиты от спама:

- для открытых мультимедийных данных без шифрования (см. вариант 1 ниже); или
- в комбинации с зашифрованными мультимедийными данными в том случае, когда **EncryptionCapability** указывает на алгоритм шифрования (см. вариант 2 ниже).

Оба варианта обеспечивают упрощенную **аутентификацию RTP пакетов** в выбранных полях посредством вычисленного кода аутентификации сообщений (MAC). MAC может быть вычислен с помощью идентификаторов объектов, определенных в 11.2.1. Криптографические алгоритмы основываются на:

- алгоритме шифрования (например, DES в режиме MAC, см. ИСО/МЭК 9797). На использование DES-MAC указывает OID "N", тогда как идентификатор объекта "O" свидетельствует о применении тройного DES-MAC; или
- использовании криптографической односторонней функции (например, SHA1). Используемый OID равен "M".

В идентификаторе объекта **antiSpamAlgorithm** также указывается на применение алгоритма MAC. В неявном виде идентификатор объекта алгоритма также указывает на размер MAC; например, 1 блок = 64 битам для DES MAC. Для экономии ширины полосы MAC можно сократить, хотя при этом в некоторой степени приходится жертвовать защитой, например, до 32-битового MAC; для этого требуется другой идентификатор объекта. Метод защиты от спама не зависит от какого-либо дополнительного шифрования полезной нагрузки (см. варианты 1 и 2 ниже).

При защите от спама используется приведенный ниже формат RTP пакетов (см. рисунок 7), где последовательность заполнения RTP интерпретируется следующим (см. A.5/H.225.0).

- Бит P в заголовке RTP должен быть установлен в 1.
- Заполняющие байты должны добавляться в конце полезной нагрузки при следующем значении:

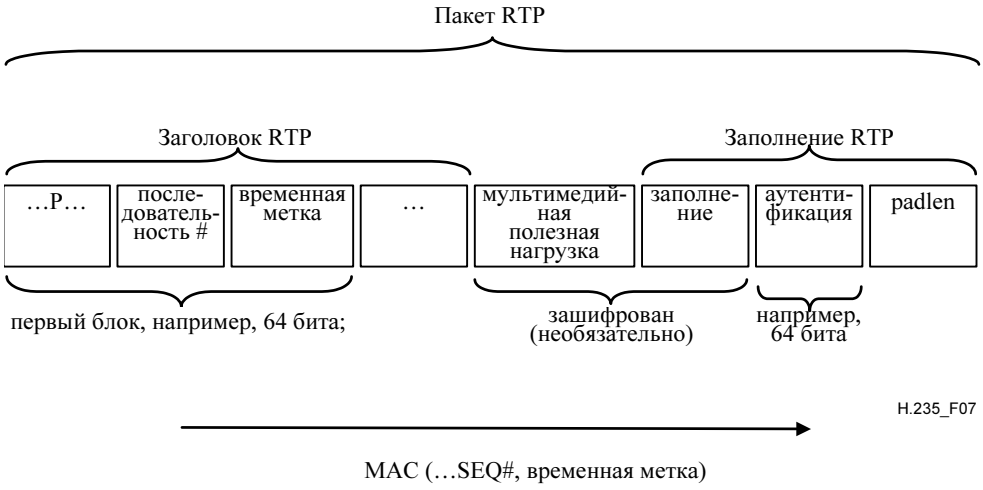


Рисунок 7/H.235 – Формат пакетов RTP для защиты от спама при передаче мультимедийной информации

ПРИМЕЧАНИЕ 1. – Если средства защиты от спама не используются, то поля AUTH и padlen тоже не используются, и применяется обычный формат пакета RTP.

1) *Случай применения средств исключительно для защиты от спама*

Этот вариант касается ситуации, при которой мультимедийные данные не зашифрованы, и поля заполнения остаются пустыми. Последний октет заполнения RTP содержит сумму тех октетов

заполнения, которыми следует пренебречь в конце пакета RTP. Остальные байты заполнения передают MAC. MAC должен вычисляться по первому криптоблоку заголовка RTP, включающего переменную временную метку и номер последовательности, посредством использования согласованного алгоритма MAC **antiSpamAlgorithm** и симметричного "ключа". Статический или конфигурируемый вручную общий "ключ" или динамически согласованный общий "ключ" *k* могут использоваться согласно процедурам Рек. МСЭ-Т H.235. Для блоков большего размера (более 64 битов) надо брать некоторые значащие дополнительные биты заголовка RTP или даже первые биты мультимедийной полезной нагрузки.

В качестве ключа для расчета MAC рекомендуется использовать ключ, получаемый в результате распределения мультимедийных сеансовых ключей H.235, хотя применяемый сеансовый ключ не употребляется для шифрования полезной нагрузки. Для управления ключами может использоваться защищенное быстрое соединение с созданием ключа (см. Приложение J/H.323) или ручные манипуляции с ключом. Отправитель вычисляет MAC в соответствии с данным выше описанием и включает результат в поле MAC – внутри поля AUTH заполнения RTP. Отправитель и получатель узнают размер поля AUTH и длину MAC посредством **antiSpamAlgorithm**.

Проверка MAC на принимающей стороне должна осуществляться как можно раньше, по возможности уже внутри стека RTP или, самое позднее, перед расшифровкой или декомпрессией полезной нагрузки. Получатель сначала повторно вычисляет MAC таким же образом, как это делал отправитель, и сравнивает полученный MAC с доставленным MAC в заполнении RTP. Если эти два MAC не соответствуют друг другу, это значит, что заголовок RTP был изменен при переходе или был отправлен неправомочным объектом, у которого нет этого ключа. Таким образом, неаутентифицированный пакет RTP должен быть сброшен, данное событие может быть зарегистрировано; вероятно, это свидетельствует о попытке нарушения защиты типа "отказ в обслуживании". В ином случае, может проводиться дальнейшая обработка аутентифицированного RTP пакета, заполнение RTP убирается, и полезная нагрузка подается через кодек.

ПРИМЕЧАНИЕ 2. – Упрощенное вычисление/верификация MAC при шифровании DES включает в себя только одну операцию шифрования; в ином случае, SHA1 MAC вычисляются исходя из сегмента пакетов фиксированной длины, следовательно, криптографические операции минимально потребляют ресурсы обработки данных.

2) *Случай применения средств для защиты от спама и шифрования полезной нагрузки*

Этот вариант относится к ситуации, при которой мультимедийные данные зашифрованы и активизированы средства защиты от спама. Если полезная нагрузка не совпадает с границами четных блоков, необходимо добавлять к полезной нагрузке перед MAC некоторые дополнительные заполняющие байты. Шифрование мультимедийной полезной нагрузки осуществляется согласно настоящему пункту 11.

EncryptionCapability определяет алгоритм шифрования полезной нагрузки, а **antiSpamAlgorithm** – метод защиты от спама. В целях обеспечения защиты, для шифрования мультимедийной информации и для MAC должны использоваться разные сеансовые ключи. Для вычисления ключа MAC *k* ключ шифрования *K* вводится посредством односторонней функции хеширования SHA1;

k = SHA1(K); значащие биты должны быть взяты из результата хеширования в порядке прихода байтов по сети. В том случае, когда **antiSpamAlgorithm** указывает на алгоритм шифрования, из собранных битов должен быть сформирован правильный ключ шифрования (например, установка контрольных битов четности DES).

После того, как получатель успешно проверит аутентичность пакета RTP, полезная нагрузка расшифровывается, а затем заполнение RTP сбрасывается. Общая процедура соответствует варианту 1, описанному выше.

11.2.1 Список идентификаторов объектов

В таблице 3 перечислены все идентификаторы объектов (OID), на которые делаются ссылки.

Таблица 3/H.235 – Идентификаторы объектов, используемые для защиты от спама

Опорное значение идентификатора объекта	Значение идентификатора объекта	Описание
"M"	{itu-t (0) recommendation (0) h (8) 235 version (0) 2 8}	Защита от спама с применением HMAC-SHA1-96
"N"	{iso(1), identified-organization(3), oiw(14), secsig(3), algorithm(2), desMAC(10)}	Защита от спама с применением DES (56 битов) MAC (см. ИСО/МЭК 9797) при 64-битовом MAC
"O"	{iso(1) identified-organization(3) oiw(14) secsig(3) algorithm(2) desEDE(17)}	Защита от спама с применением тройного-DES (168-битов) MAC (см. ИСО/МЭК 9797)

12 Устранение ошибок защиты

В настоящей Рекомендации не определяются и не рекомендуются никакие методы, с помощью которых конечные точки могут контролировать свою абсолютную секретность. В ней, однако, рекомендуются действия, которые следует предпринимать при обнаружении потери секретности.

Если какая-либо конечная точка обнаруживает нарушение защиты в канале установления соединения (например, H.225.0 для H.323), то она должна немедленно прекратить соединение, следуя процедурам протоколов, соответствующим данной конечной точке (согласно 8.5/H.323 за исключением этапа B-5).

Если какая-либо конечная точка обнаруживает нарушение защиты канала H.245 или логического канала передачи защищенных данных (**h235Control**), то она должна немедленно блокировать соединение, следуя процедурам протоколов, соответствующим данной конечной точке (согласно 8.5/H.323 за исключением этапа B-5).

Если любая конечная точка обнаруживает потерю секретности в одном из логических каналов, то она должна немедленно запросить новый ключ (**encryptionUpdateRequest**) и/или закрыть этот логический канал. При выявленной MC(U) потери секретности в одном логическом канале, это может вызвать закрытие всех остальных логических каналов и/или изменение их ключей, по усмотрению MC(U). MC(U) должен будет направить **encryptionUpdateRequest**, **encryptionUpdate** ко всем конечным точкам, которых это касается.

По усмотрению MC(U), ошибка защиты в отдельном канале может вызвать блокирование соединений со всеми конечными точками, участвующими в конференции, и, таким образом, привести к завершению конференции.

13 Асимметричная аутентификация и обмен ключами с использованием систем шифрования методом эллиптических кривых

В настоящей Рекомендации представлены сложные методы шифрования, основанные на эллиптических кривых и применяемые для подписей, управления ключами и шифрования. Некоторые их основные преимущества над "классическими" асимметричными методами шифрования, такими как RSA, следующие:

- Более короткие криптографические ключи, создающие защиту, сравнимую с защитой при RSA: Типичная длина ключа для криптосистем с шифрованием методом эллиптических кривых составляет 160 битов, что, с точки зрения защиты, эквивалентно 1024-битовому ключу RSA. Для хранения более короткого ключа требуется меньший объем памяти; его применение делает криптосистемы с шифрованием методом эллиптических кривых особенно привлекательными для реализации в смарт-картах и любых других устройствах с низкими требованиями к объему памяти. В условиях H.323 конечные точки более простого типа – для передачи аудиосигналов с защитой (SASET), представленные в Приложении J/H.323, которые имеют более низкие требования в части цены, хорошо подходят для применения шифрования методом эллиптических кривых.
- Повышение скорости обработки, которое достигается за счет реализации, как программного обеспечения, так и аппаратного обеспечения. Применение более коротких ключей способствует ускорению обработки. Это приводит к более быстрым интерактивным откликам (пользователей).

Все основные сведения, пояснения и процедуры обработки при шифровании методом эллиптических кривых можно найти в (*Спецификации обеспечения защиты ATM Версия 1.1*, раздел 8.7). Рекомендуется

шифровать эллиптические точки в их аффинной несжатой нотации без использования метода точечной компрессии/декомпрессии. Дополнительную информацию по этой теме можно найти в Рекомендациях ИСО/МЭК 15946-1 и ИСО/МЭК 15946-2.

13.1 Управление ключами

Схемы согласования ключей Диффи-Хеллмана, шифрованных методом эллиптических кривых подобны классическому случаю mod-p, также определенному в настоящей Рекомендации. Имеется два варианта:

- эллиптические кривые в исходном поле: **eckasdhp** содержит параметры эллиптической кривой и Диффи-Хеллмана;
- эллиптические кривые характеристики 2: **eckasdh2** содержит параметры эллиптической кривой и Диффи-Хеллмана.

Структура ECKASDH используется для любого варианта. Некоторые примеры эллиптических кривых приведены в Рек. ИСО/МЭК 15946-1. Также можно использовать любые другие подходящие и приемлемые эллиптические кривые.

Из-за наличия упорядоченной структуры **ClearToken**, передача **dhkey** и **eckasdhkey** не должна происходить одновременно; при осуществлении обмена ключами Диффи-Хеллмана должна присутствовать только одна из них.

Замечание. – Не следует путать произвольно выбранные секретные параметры ***a*** стороны A или ***b*** стороны B с распространенными коэффициентами Вейерштрасса ***a, b***.

13.2 Цифровая подпись

В поле **ECGDSASignature** передаются значения **r** и **s** вычисленной цифровой подписи, базирующейся на шифровании методом эллиптических кривых. Дополнительная информация об алгоритме подписи EC-GDSA приведена в Разделе 8.7.3 *Спецификации обеспечения защиты АТМ Версия 1.1* и в главе 5 документа ИСО 15946-2.

Цифровая подпись **ECGDSA**, базирующаяся на шифровании методом эллиптических кривых, должна кодироваться в ASN.1 и затем вставляться в поле **signature** макрокоманды **SIGNED** настоящей Рекомендации. Для цифровой подписи отправитель должен будет включить идентификатор объекта в **algorithmOID**, с помощью которого получатель csможет определить использование цифровой подписи, шифрованной методом эллиптических кривых.

Приложение A

H.235 ASN.1

```
H235-SECURITY-MESSAGES DEFINITIONS AUTOMATIC TAGS ::=
BEGIN

-- ЭКСПОРТИРУЕТ ВСЕ

ChallengeString         ::= OCTET STRING (SIZE (8..128))
TimeStamp               ::= INTEGER(1..4294967295)     -- секунды 00:00
                                                       -- 1/1/1970 по Гринвичу
RandomVal               ::= INTEGER -- 32-битовое целое
Password                ::= BMPString (SIZE (1..128))
Identifier              ::= BMPString (SIZE (1..128))
KeyMaterial             ::= BIT STRING(SIZE(1..2048))

NonStandardParameter ::= SEQUENCE
{
     nonStandardIdentifier   OBJECT IDENTIFIER,
     data                    OCTET STRING
}
```

```
-- если применяется локальное октетное представление этих строк битов, то они
-- должны использовать стандартный порядок сетевых октетов (например, Big
Endian)
DHset ::= SEQUENCE
{
     halfkey       BIT STRING (SIZE(0..2048)), -- = g^x mod n
     modSize       BIT STRING (SIZE(0..2048)), -- n
     generator     BIT STRING (SIZE(0..2048)), -- g
     ...
}

ECpoint ::= SEQUENCE -- представление точки эллиптической кривой в аффинных
                     -- координатах без компрессии (x, y)
{
     x         BIT STRING (SIZE(0..511)) OPTIONAL,
     y         BIT STRING (SIZE(0..511)) OPTIONAL,
     ...
}

ECKASDH::= CHOICE  -- параметры для схемы Диффи-Хеллмана согласования ключей на
                   -- основе эллиптической кривой
{
     eckasdhp SEQUENCE -- параметры для эллиптических кривых исходного поля
     {
      public-key        ECpoint, -- Данное поле включает представление значения
               -- открытого ключа ECKAS-DHp.
               -- Это поле включает значение открытого ключа(aP) ECKAS-DHp, когда
               -- этот элемент информации передается от инициатора к получателю.
               -- Данное поле включает  значение открытого ключа (bP) ECKAS-DHp,
               -- когда этот элемент информации передается обратно от получателя
               -- к инициатору.
          modulus        BIT STRING (SIZE(0..511)), -- Данное поле включает
               -- представление открытой величины модуля (p) ECKAS-DHp.
          base           ECpoint, -- Данное поле включает представление открытой
               -- базы (P) ECKAS-DHp.
          weierstrassA  BIT STRING (SIZE(0..511)), -- Данное поле включает
               -- представление коэффициента Вейерштрасса(a)ECKAS-DHp.
          weierstrassB  BIT STRING (SIZE(0..511)) -- Данное поле включает
               -- представление коэффициента Вейерштрасса(b)ECKAS-DHp.
     },

     eckasdh2 SEQUENCE -- параметры для эллиптических кривых характеристики 2
     {
          public-key    ECpoint, -- Данное поле включает представление
               -- значения открытого ключа ECKAS-DH2.
               -- Это поле включает значение открытого ключа(aP)ECKAS-DH2, когда
               -- этот элемент информации передается от инициатора к получателю.
               -- Данное поле включает значение открытого ключа(bP)ECKAS-DH2,
               -- когда этот элемент информации передается обратно от получателя
               -- к инициатору.
          fieldSize     BIT STRING (SIZE(0..511)), -- Это поле включает
               -- представление величины размера поля (m) ECKAS-DH2.
          base          ECpoint, -- Данное поле включает представление открытой
               -- базы(P) ECKAS-DH2.
          weierstrassA  BIT STRING (SIZE(0..511)), -- Данное поле включает
               -- представление коэффициента Вейерштрасса(a) ECKAS-DH2.
          weierstrassB  BIT STRING (SIZE(0..511)) -- Данное поле включает
               -- представление коэффициента Вейерштрасса(b)ECKAS-DH2.
     },
     ...
}
```

```
ECGDSASignature ::= SEQUENCE -- параметры алгоритма цифровых подписей,
              -- шифрованных методом эллиптических кривых
{
    r         BIT STRING (SIZE(0..511)), -- Данное поле включает
              -- представлениекомпонента r цифровой подписи ECGDSA.
    s         BIT STRING (SIZE(0..511)) -- Данное поле включает
              -- представление компонента s цифровой подписи ECGDSA.
}

TypedCertificate ::= SEQUENCE
{
    type          OBJECT IDENTIFIER,
    certificate   OCTET STRING,
    ...
}

AuthenticationBES ::= CHOICE
{
    default       NULL, -- зашифрованный ClearToken
    radius        NULL, -- запрос/подтверждение RADIUS
    ...
}

AuthenticationMechanism     ::= CHOICE
{
    dhExch        NULL, -- Диффи-Хеллман
    pwdSymEnc     NULL, -- пароль с симметричным шифрованием
    pwdHash       NULL, -- пароль с хешированием
    certSign      NULL, -- Сертификат с подписью
    ipsec         NULL, -- соединение на основе IPSEC
    tls           NULL,
    nonStandard   NonStandardParameter, -- что-то иное.
    ...,
    authenticationBES AuthenticationBES -- аутентификация пользователя для
                                        -- BES
}

ClearToken        ::= SEQUENCE  -- "маркер" может содержать множество типов
                                -- значений.
{
    tokenOID      OBJECT IDENTIFIER,
    timeStamp         TimeStamp OPTIONAL,
    password      Password OPTIONAL,
    dhkey         DHset OPTIONAL,
    challenge     ChallengeString OPTIONAL,
    random        RandomVal OPTIONAL,
    certificate   TypedCertificate OPTIONAL,
    generalID     Identifier OPTIONAL,
    nonStandard   NonStandardParameter OPTIONAL,
    ...,
    eckasdhkey    ECKASDH OPTIONAL, -- аналоговый вариант
                                    -- схемы согласования ключа Диффи-
                                    -- Хеллмана, шифрованных методом
                                    -- эллиптических кривых (ECKAS-DH)
    sendersID     Identifier OPTIONAL,
    h235Key       H235Key OPTIONAL -- центральный распределенный ключ в V3
}

--  Идентификатор объекта должен быть помещен в поле  tokenOID, когда
--  ClearToken включается непосредственно в сообщение (в отличие от
--  зашифрованного варианта). Во всех других случаях приложение должно
--  использовать идентификатор объекта { 0 0 }, чтобы показать, что
--  значение tokenOID не присутствует.
```

```
--   Запускает все криптографически параметризованные типы здесь...
--

SIGNED { ToBeSigned } ::= SEQUENCE {
     toBeSigned          ToBeSigned,
     algorithmOID        OBJECT IDENTIFIER,
     paramS              Params,   -- любые параметры "рабочего цикла"
     signature           BIT STRING  -- может быть подписью ECGDSA, зашифрованной
в RSA или в ASN.1
} ( CONSTRAINED BY { -- Проверить или подписать сертификат -- } )


ENCRYPTED { ToBeEncrypted } ::= SEQUENCE {
     algorithmOID        OBJECT IDENTIFIER,
     paramS              Params,   -- любые параметры "рабочего цикла"
     encryptedData       OCTET STRING
} ( CONSTRAINED BY { -- Шифровать или расшифровать -- ToBeEncrypted } )

HASHED { ToBeHashed } ::= SEQUENCE {
     algorithmOID        OBJECT IDENTIFIER,
     paramS              Params,   -- любые параметры "рабочего цикла"
     hash                BIT STRING
} ( CONSTRAINED BY { -- ••• -- ToBeHashed } )

IV8 ::= OCTET STRING (SIZE(8))    -- начальное значение для 64-битовых блочных
                                  -- шифров
IV16 ::= OCTET STRING (SIZE(16)) -- начальное значение для 128-битовых блочных
                                  -- шифров

-- Используемый алгоритм подписи должен выбирать один из этих типов параметров,
-- требуемых стороной, принимающей подпись.

Params ::= SEQUENCE {
     ranInt         INTEGER OPTIONAL, -- некоторая целая величина
     iv8            IV8 OPTIONAL, -- вектор инициализации, состоящий из 8 октетов
     ...,
     iv16           IV16 OPTIONAL,-- вектор инициализации, состоящий из 16
                                  --октетов
     iv             OCTET STRING OPTIONAL,  -- произвольная инициализация длины
     вектора
     clearSalt   OCTET STRING OPTIONAL -- нешифрованный ключ "с привязками" для
     кодирования
}

EncodedGeneralToken ::= TYPE-IDENTIFIER.&Type (ClearToken -- маркер общего
назначения -- )
PwdCertToken ::= ClearToken (WITH COMPONENTS {..., timeStamp PRESENT, generalID
PRESENT})
EncodedPwdCertToken ::= TYPE-IDENTIFIER.&Type (PwdCertToken)

CryptoToken::= CHOICE
{

     cryptoEncryptedToken SEQUENCE -- Маркер общего назначения/конкретного
     приложения
     {
          tokenOID       OBJECT IDENTIFIER,
          token          ENCRYPTED { EncodedGeneralToken }
     },
     cryptoSignedToken  SEQUENCE -- Маркер общего назначения/конкретного
     приложения
     {
          tokenOID       OBJECT IDENTIFIER,
          token          SIGNED { EncodedGeneralToken }
     },
     cryptoHashedToken SEQUENCE -- Маркер общего назначения/конкретного
     приложения
```

```
    {
        tokenOID            OBJECT IDENTIFIER,
        hashedVals          ClearToken,
        token HASHED { EncodedGeneralToken }
    },
    cryptoPwdEncr ENCRYPTED { EncodedPwdCertToken },
    ...
}

-- Это обеспечивает прохождение сеансовых ключей внутри структуры OLC H.245.
-- Они зашифрованы как самостоятельные элементы в ASN.1 и основываются на OCTET
-- STRING в H.245
H235Key  ::= CHOICE -- он используется с полем ClearToken "h235Key" или с H.245
{
    secureChannel           KeyMaterial,
    sharedSecret            ENCRYPTED {EncodedKeySyncMaterial},
    certProtectedKey        SIGNED { EncodedKeySignedMaterial },
    ...,
    secureSharedSecret      V3KeySyncMaterial -- ••• •••••••• ••••• V3 H.235
}

KeySignedMaterial ::= SEQUENCE {
    generalId     Identifier, -- псевдоним ведомого терминала
    mrandom       RandomVal, -- случайное значение ведущего терминала
    srandom       RandomVal OPTIONAL, -- случайное значение ведомого терминала
    timeStamp     TimeStamp OPTIONAL, -- временная метка ведущего терминала для
                                      -- незапрашиваемого исполнительного
                                      -- устройства (EU)
    encrptval     ENCRYPTED {EncodedKeySyncMaterial }
}
EncodedKeySignedMaterial ::= TYPE-IDENTIFIER.&Type (KeySignedMaterial)

H235CertificateSignature    ::= SEQUENCE
{
    certificate             TypedCertificate,
    responseRandom          RandomVal,
    requesterRandom         RandomVal OPTIONAL,
    signature               SIGNED { EncodedReturnSig },
    ...
}

ReturnSig ::= SEQUENCE {
    generalId          Identifier, -- псевдоним ведомого терминала
    responseRandom     RandomVal,
    requestRandom      RandomVal OPTIONAL,
    certificate        TypedCertificate OPTIONAL -- затребованный сертификат
}

EncodedReturnSig ::= TYPE-IDENTIFIER.&Type (ReturnSig)
KeySyncMaterial    ::= SEQUENCE
{
    generalID     Identifier,
    keyMaterial   KeyMaterial,
    ...
}
EncodedKeySyncMaterial  ::= TYPE-IDENTIFIER.&Type (KeySyncMaterial)



V3KeySyncMaterial  ::= SEQUENCE
{
    generalID               Identifier OPTIONAL, -- ID однорангового терминала
    algorithmOID            OBJECT IDENTIFIER OPTIONAL, -- алгоритм шифрования
    paramS                  Params, -- IV
    encryptedSessionKey     OCTET STRING OPTIONAL, -- зашифрованный сеансовый
                                        -- ключ
```

```
    encryptedSaltingKey     OCTET STRING OPTIONAL, -- шифрованный мультимедийный
                                                   -- ключ с "привязками"
    clearSaltingKey         OCTET STRING OPTIONAL, -- нешифрованный
                                                   -- мультимедийный ключ с "привязками"
    paramSsalt              Params OPTIONAL, -- IV (и нешифрованной "привязки")
                                                   -- для шифрования ключа с "привязками"
    keyDerivationOID        OBJECT IDENTIFIER OPTIONAL, -- метод формирования
                                                   -- ключей
    ...
}

END   -- Окончание ОПРЕДЕЛЕНИЙ СООБЩЕНИЙ ЗАЩИТЫ H.235
```

Приложение B

Вопросы, касающиеся рекомендации H.323

B.1 Общие сведения

На рисунке B.1 представлены вопросы, рассматриваемые в настоящей Рекомендации, применительно к Рек. МСЭ-Т H.323.

Рисунок B.1/H.235 – Обзор вопросов

Для Рек. МСЭ-Т H.323 передача сигналов об использовании протоколов TLS, IPSEC или какого-либо собственного механизма в канале управления H.245.0 должна производиться по защищенному или незащищенному каналу H.225.0 во время первоначального обмена сообщениями Q.931.

B.2 Передача сигналов и соответствующие процедуры

Должны выполняться процедуры, изложенные в пункте 8/H.323 ("Процедуры передачи сигналов вызова"). Конечные точки H.323 должны иметь возможность шифровать и распознавать наличие (или отсутствие) требований защиты (для канала H.245), передаваемых в сообщениях H.225.0.

В случае, когда нужно обеспечить защиту самого канала H.225.0, должны выполняться те же самые процедуры из пункта 8/H.323. Различие в работе состоит в том, что процесс связи должен будет

происходить только после подключения к защищенному идентификатору TSAP и использования заранее определенных режимов защиты (например, TLS). В связи с тем, что при организации процесса связи согласно H.323 сначала производится обмен сообщениями H.225.0, не может быть никакого "внутриполосного" согласования защиты "в полосе" для H.225.0. Иными словами, обе стороны должны *заранее* знать, что они используют конкретный режим защиты. При H.323 в IP для осуществления процесса связи под защитой TLS используется альтернативный общеизвестный порт (1300).

Одна из целей обмена сообщениями H.225.0 в соответствии с защитой H.323 – это создание механизма для организации защищенного канала H.245. По выбору аутентификация может происходить при обмене сообщениями H.225.0. Эта аутентификация может основываться на сертификате или пароле, использовать шифрование и/или хеширование (т. е. осуществление подписания). Специфика этих режимов работы описана в пунктах 10.2–10.3.4.

Конечная точка H.323, которая принимает сообщение SETUP с набором **h245SecurityCapability**, должна будет ответить соответствующим приемлемым **h245SecurityMode** в сообщении CONNECT. В тех случаях, когда отсутствуют наложенные характеристики, вызываемый терминал может отказать в соединении, отправив сообщение **Release Complete** с кодом причины, установленным в ***SecurityDenied***. Сообщение об ошибке не предназначено для передачи какой-либо информации относительно несоответствия защиты; вызывающему терминалу придется каким-либо иным образом определять, в чем состоит проблема. Если вызывающий терминал получает сообщение CONNECT при отсутствии достаточного или приемлемого режима защиты, он может завершить вызов посредством **Release Complete** с ***SecurityDenied***. В тех случаях, когда вызывающий терминал получает сообщение CONNECT без каких-либо защитных характеристик, он может завершить вызов посредством **Release Complete** с ***undefinedReason***.

Если вызывающий терминал получает сообщение о приемлемом режиме **h245Security**, то он должен открыть и использовать канал H.245 в указанном режиме защиты. Безуспешное установление канала H.245 в определенный здесь режим защиты следует рассматривать как ошибку протокола, и соединение блокируется.

B.2.1 Совместимость с Редакцией 1

Конечная точка, имеющая возможности защиты, не должна возвращать конечной точке, не обладающей возможностями защиты, никакие, относящиеся к защите поля, указания или статус. Если вызывающая сторона получает сообщение SETUP, которое не содержит характеристик **h245Security** и/или маркер аутентификации, то она может вернуть **Release Complete** для того, чтобы отказаться от установления соединения; но в этом случае она должна будет использовать код причины ***undefinedReason***. Соответственно, если, отправив сообщение SETUP с **h245Security** и/или маркером аутентификации, вызывающая сторона получает сообщение CONNECT без **h245SecurityMode** и/или маркера аутентификации, то она также может блокировать соединение, выдав **Release Complete** с кодом причины ***UndefinedReason***.

B.2.2 Передача сигнала ошибки

Имеющий возможности защиты контроллер доступа или другой объект H.225.0, имеющий расширенные возможности защиты, должны обеспечивать индикацию ошибки. Ошибка защиты свидетельствует о том, что этот объект не смог правильно обработать полученное сообщение. Там, где только возможно, необходимо обеспечивать наличие детального кода ошибки.

- **securityWrongSyncTime** будет означать, что отправитель выявил проблему защиты в виде несоответствующих временных меток. Это может быть вызвано проблемами с временным сервером, потерей синхронизации или это может быть следствием чрезмерного запаздывания сети.
- **securityReplay** будет указывать на то, что выявлена повторная попытка нарушения защиты. Это происходит в том случае, когда один и тот же номер последовательности встречается несколько раз в конкретной временной метке.
- **securityWrongGeneralID** будет указывать на несоответствие общего ID в этом сообщении. Это может быть вызвано неправильной адресацией.
- **securityWrongSendersID** будет указывать на несоответствие ID отправителя в этом сообщении. Это может быть вызвано ошибочностью вводимых данных пользователя.
- **securityIntegrityFailed** будет указывать на неуспешное выполнение проверки целостности/подписи. Согласно Приложению D, это может быть вызвано неправильным или неправильно определенным паролем во время первоначального запроса или вследствие

выявленной активной попытки нарушения защиты. Согласно Приложениям E/F, это сообщение будет указывать на неуспешное выполнение проверки цифровой подписи в сообщении. Это может быть вызвано неправильным открытым/личным ключом или выявленной активной попыткой нарушения защиты.

- **securityWrongOID** будет указывать на любое несоответствие в OID меток (шифрованная или нешифрованная метка) или OID криптографического алгоритма. Указывает на различные реализованные профили/алгоритмы защиты.
- **securityDHmismatch** будет указывать на любое несоответствие обмениваемых параметров Диффи-Хеллмана. Может указывать на различные наборы параметров DH или даже на различные реализованные алгоритмы шифрования речевых сигналов.
- **securityCertificateExpired** будет указывать на то, что срок действия сертификата истек.
- **securityCertificateDateInvalid** будет указывать на то, что сертификат все еще недействителен.
- **securityCertificateRevoked** будет указывать на то, что сертификат оказался отмененным.
- **securityCertificateNotReadable** будет указывать на то, что сертификат не может быть правильно декодирован в ASN.1 или он имеет другую несоответствующую форму.
- **securityCertificateSignatureInvalid** будет указывать на то, что подпись в сертификате неправильная.
- **securityCertificateMissing** будет указывать на то, что, как оказалось, ожидаемый сертификат отсутствует или что этот сертификат не может быть локализован иным образом.
- **securityCertificateIncomplete** будет указывать на то, что некоторые ожидаемые добавления к сертификату отсутствуют.
- **securityUnsupportedCertificateAlgOID** будет указывать на то, что определенные криптографические алгоритмы, такие, как хеширование или цифровые подписи, используемые в рамках сертификата, не поняты или не обеспечиваются. В качестве составляющей ответа возврата, отправитель может представлять перечень допустимых сертификатов в отдельных маркерах с тем, чтобы содействовать выбору получателем соответствующего варианта.
- **securityUnknownCA** будет указывать на то, что корневой сертификат/сертификат CA не могут быть выявлены или же, что этот сертификат невозможно согласовать с доверительным CA.

В любом другом случае, если команда защиты H.235 выполнена неуспешно, должен быть возвращен **securityDenial** для RAS H.225.0 (соответственно, **securityDenied** для передачи сигналов вызова H.225.0).

ПРИМЕЧАНИЕ 1. – Среди профилей защиты в Приложении D, Приложении E или Приложении F могут встретиться securityWrongSyncTime, securityReplay, securityWrongGeneralID, securityWrongSendersID, SecurityIntegrityFailed, securityDHmismatch и securityWrongOID.

ПРИМЕЧАНИЕ 2. – Среди профилей защиты в Приложении E или Приложении F могут встретиться – securityCertificateExpired, securityCertificateDateInvalid, securityCertificateRevoked, securityCertificateNotReadable, securityCertificateSignatureInvalid, securityCertificateMissing, securityCertificateIncomplete, securityUnsupported CertificateAlgOID и securityUnknownCA.

B.2.3 Определение возможностей версии 3

Конечные точки H.235, версии 3 и более поздних версий обеспечивают усовершенствованные процедуры защиты мультимедийного тракта, что не скажешь о Рекомендации H.235, версии 1 и 2. К этим усовершенствованным процедурам защиты относятся:

- усовершенствованная транспортировка ключа (**V3KeySyncMaterial**, см. B.2.4.1);
- усовершенствованное обновление ключа, см. B.2.6.2.

Так как конечные точки обычно не осведомлены о взаимной поддержке H.235, версии 3, то во время установления соединения добавляется точная индикация версии.

Конечные точки H.235, версии 3 и более поздних версий всегда должны использовать процедуры, описанные в этом разделе, для определения возможностей версии 3 (усовершенствованная транспортировка ключа, усовершенствованная синхронизация шифрования). В зависимости от результатов процедуры передачи логических сигналов, конечные точки могут использовать эти процедуры (см. B.2.4) для обратной совместимости с конечными точками H.235, версии 1 или версии 2.

Для указания о том, использовать ли усовершенствованные процедуры H.235, версии 3, вызывающая и вызываемая конечные точки должны во время передачи сигналов вызова (SETUP, CONNECT и т. д.) включать дополнительный **ClearToken**, указывающий на возможности версии 3. Отсутствие такого **ClearToken** будет указывать на поддержку возможностей только H.235, версии 1 или версии 2. В этом случае конечная точка должна применять процедуру в B.2.4. В ином случае, конечная точка может применять усовершенствованные процедуры, как описано в B.2.4.1, или применять процедуру H.235, версии 1 или версии 2 в B.2.4.

Этот **ClearToken** должен будет применять **tokenOID**, установленный "V3", и ему присваивается следующее значение.

"V3"	{itu-t (0) recommendation (0) h (8) 235 version (0) 3 24}	Индикатор возможностей версии 3 в ClearToken во время передачи сигналов вызова.

Любые другие поля в этом **ClearToken** должны оставаться неиспользованными, если только они не используются для передачи параметров DH.

B.2.4 Транспортировка ключей

Ведущий объект должен формировать данные сеансового ключа и распространять их к одноранговому объекту(ам). Для транспортировки ключа предлагается две процедуры:

- процедура, рассчитанная, в основном, на конечные точки H.235 версии 1 или версии 2; описанная в данном разделе;
- усовершенствованная процедура, рассчитанная на конечные точки H.235 версии 3 описанная в B.2.4.1.

Конечные точки H.235 версии 1 или версии 2 применяют следующую процедуру для транспортировки сеансовых ключей:

KeySyncMaterial содержит идентификатор конечной точки ведущего объекта в рамках сообщения **generalID** и переносит данные сеансового ключа в рамках **keyMaterial**. Значение **generalID** должно быть включено для обеспечения минимального уровня аутентификации источника сеансового ключа (см. также D.7.2). Получатель должен будет проверять правильность полученного **generalID**.

ПРИМЕЧАНИЕ. – В данной Рекомендации предполагается, что каждая конечная точка зарегистрирована контроллером доступа, и ей присвоен идентификатор конечной точки, который может быть передан в рамках **generalID**. В данной Рекомендации не представлены сценарии без контроллера доступа; этот вопрос подлежит дальнейшему изучению.

KeySyncMaterial должен быть зашифрован с использованием согласованного основного ключа. Перед шифрованием к множеству блоков всегда должен добавляться элемент **KeySyncMaterial**, при этом последнему октету должно быть задано значение, соответствующее количеству октетов – заполнителей (включая последний). Значение заполнителя должно определяться посредством обычного согласования алгоритма шифрования. Результат шифрования должен будет храниться в **sharedSecret** ключа **H235Key**.

B.2.4.1 Усовершенствованная транспортировка ключа согласно H.235, версии 3

Было замечено, что описанный в синтаксисе ASN.1 элемент **KeySyncMaterial** и метод применения к данным команды ENCRYPTED{} в H.235, версии 1 и 2 выявляют множество известных нешифрованных текстов: прежде всего, **generalID** ведущего объекта, но также и некоторые известные биты кодирования для этой структуры. **generalID**, даже будучи зашифрованным, известен исходя из других нешифрованных частей сигнального сообщения (к примеру, **senderID**). Полагают, что присутствие таких известных нешифрованных текстов значительно ослабляет систему защиты таким образом, что объект, пытающийся нарушить защиту, сможет с большой легкостью "взломать" сеансовый ключ, применив "грубую силу", особенно при блочном шифре, имеющем меньшую длину блоков, таком как DES-56 или RC2-совместимом.

Кроме того, H.235, версии 3 должна обладать возможностями для транспортировки дополнительных данных ключа:

- Обеспечить защиту транспортировки ключа с "привязками" (salting key) к одноранговому объекту(ам). Такой ключ вводится для усовершенствованного режима OFB; см. B.2.5.

В H.235, версии 3 элемент **H235Key** дополняется элементом **secureSharedSecret**, содержащим **V3KeySyncMaterial**, который включает следующие параметры:

generalID содержит идентификатор конечной точки инициирующего отправителя, если он имеется, в противном случае, это поле остается неиспользованным.

algorithmOID указывает на применяемый криптографический алгоритм и режим работы.

paramS содержит значение инициализации, которое применяется для шифрования передаваемого ключа(ей).

ПРИМЕЧАНИЕ 1. – Значение IV в рамках **paramS** не надо путать с величиной IV каждого пакета RTP, который не передается. **ClearSalt** дополнительно содержит нешифрованный ключ с "привязками" (salting key) для шифрования сеансовых ключей (к примеру, EOFB).

encryptedSessionKey содержит шифрованный текст зашифрованного исходного сеансового ключа.

encryptedSaltingKey содержит шифрованный текст зашифрованного ключа с "привязками" (salting key), если он вообще существует. Этот ключ необходим для усовершенствованного режима OFB.

clearSaltingKey может содержать незашифрованный исходный мультимедийный ключ с "привязками" (salting key). При применении его необходимо убедиться в том, что **encryptedSaltingKey** и **clearSaltingKey** не будут использоваться одновременно.

paramSsalt содержит исходное значение для шифрования ключа с "привязками" (salting key). **ClearSalt** факультативно содержит нешифрованный ключ с "привязками" для шифрования этого ключа (к примеру, EOFB).

ПРИМЕЧАНИЕ 2. – **generalID**, **algorithmOID** и **paramS** всегда передаются в незашифрованном тексте, тогда как **encryptedSessionKey**, **encryptedSaltingKey** содержит шифрованный текст с данными зашифрованного ключа.

Ведущий объект формирует ключ(и) в соответствии с согласованными возможностями терминала и посылает ключ(и), используя **V3KeySyncMaterial**, к одноранговой конечной точке(ам). Таким образом, **V3KeySyncMaterial**, при его наличии, должен передаваться промежуточными контроллерами доступа в неизменном виде.

Конечные точки H.235, версии 3 или последующих версий всегда должны использовать **secureSharedSecret** в рамках **H235Key**, но, в зависимости от выходных данных процедуры сигнализации логического канала в B.2.3, задействующей индицирующий **ClearToken** версии 3, они могут использовать **sharedSecret** для обратной совместимости с конечными точками H.235 версии 1 или версии 2.

B.2.5 Усовершенствованный режим OFB

Режим OFB (ИСО/МЭК 10116) определяется как режим работы, осуществляющий потоковое шифрование, используя алгоритмы блочного шифрования. Режим OFB обеспечивает:

- более совершенное качество работы посредством сокращения задержек при выполнении операции шифрования;
- более простую обработку неполных блоков;
- достаточную способность к восстановлению после возникновения ошибки по битам.

Усовершенствованный режим OFB – это слегка усовершенствованный режим OFB, названный здесь "Усовершенствованным режимом обратной связи с выхода системы" (EOFB), который имеет те же возможности, что и OFB, но, в дополнение к ним:

1) использует ключ с "привязками" (salting key) KS в дополнение к ключу шифрования KE; и
2) вводит индекс неявного пакета.

Применение дополнительного секретного ключа с "привязками" (secret salting key) KS, который подвергается логической операции "исключающее ИЛИ" в обратном направлении, создает дополнительные возможности для того, чтобы защитить систему от обработки и анализа известных нешифрованных текстов. Это основное преимущество в части защиты, которое не обеспечивают другие стандартные режимы работы (такие как CBC, OFB и т. д.). Применение режима EOFB повысит, таким образом, интенсивность защиты при обработке и анализе высокоизбыточных нешифрованных текстов, а также общеизвестных нешифрованных текстов.

Режим EOFB описывается в виде $C_i = P_i \oplus S_i$, где $S_i = E_{KE}(KS \oplus S_{i-1})$ при $i = 1 \ldots n$ и $S_0 = IV$, где C_i – это i-ный блок шифрованного текста, P_i – i-ный блок нешифрованного текста, S_i – i-ный блок потока ключей, KE – ключ шифрования, а $\oplus$ – побитовая логическая операция "исключающее ИЛИ". Режим EOFB представлен на рисунке I.4.1.

EOFB может также выполняться в стандартном режиме OFB, совмещая обратные сообщения в EOFB с сообщениями в OFB. В тех случаях, когда необходима обратная совместимость со стандартным режимом OFB, ключ шифрования с "привязками" (salting key) KS должен быть установлен во все нули или

приближенно к этому, оставляя незаполненным поле **encryptedSaltingKey** в рамках **V3KeySyncMaterial**. Тем не менее, использование реального ключа с "привязками" (salting key) весьма рекомендовано в случаях шифрования полезной нагрузки RTP посредством блочного шифра, который имеет более короткую длину блоков, такого как DES-56 или RC2-совместимый.

После обработки, по крайней мере, 2^{48} пакетов, необходимо использовать новый сеансовый ключ шифрования KE и новый ключ шифрования с "привязками" (salting key), в противном случае, произойдет повторное использование потока ключей, что ставит под угрозу безопасность данных.

В Приложении D описываются идентификаторы объектов для DES-56-EOFB, RC2-совместимого-EOFB, 3DES-EOFB и AES-EOFB.

B.2.6 Обновление ключей и синхронизация

Сеансовые мультимедийные ключи не существуют вечно. В какой-то момент времени срок действия каждого сеансового ключа истекает. Тогда для обеспечения защиты текущего сеанса надо использовать новый сеансовый ключ. В условиях конференции необходимо определить новый ключ группового сеанса связи и распределить его во время присоединения к защищенной конференции группы участников или выхода их из конференции, предупреждая, таким образом, доступ их к предыдущим или последующим данным.

- Обновление и синхронизация ключей, зависящих от типа полезной нагрузки, определяют новый тип динамической полезной нагрузки для этого нового сеансового ключа; см. B.2.6.1, B.2.6.2 и B.2.6.3.

Для обновления ключей в данной Рекомендации предлагается неподтвержденное квитирование, которое применимо для конечных точек H.235 версии 1 и версии 2, а также – устойчивое к ошибкам (robust) подтвержденное квитирование для конечных точек H.235 версии 3 и последующих версий.

B.2.6.1 Обновление неквитированных ключей

На рисунке B.1.1 представлено неподтвержденное квитирование для обновления/распределения сеансовых ключей. Если ведомому объекту требуется обновленный сеансовый ключ, то этот ведомый объект может запросить новый сеансовый ключ от ведущего объекта путем выдачи сообщения **encryptionUpdateRequest** к ведущему объекту. Этот ведущий объект должен будет послать новый сеансовый ключ (при наличии или отсутствии предварительного сообщения **encryptionUpdateRequest** от ведомого объекта) к ведомому объекту в рамках сообщения **EncryptionUpdate**.

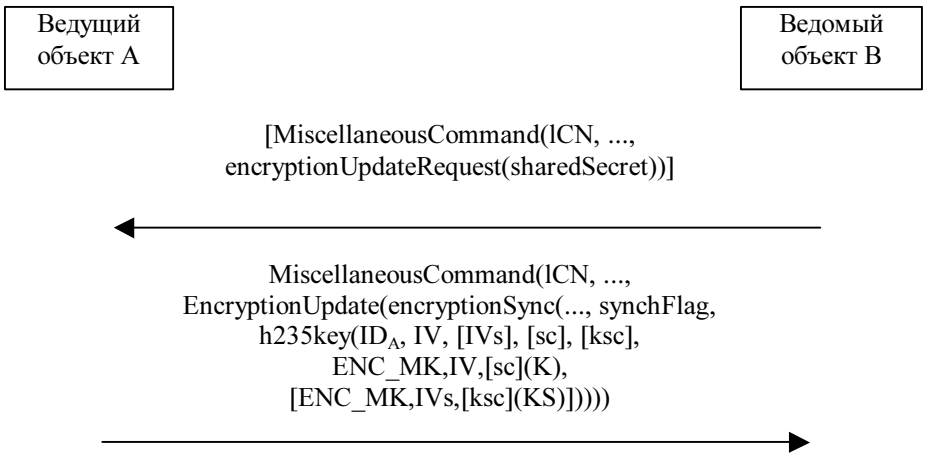


Рисунок B.1.1/H.235 – Обновление и распространение неквитированного сеансового ключа от ведущего объекта к ведомому(ым)

Где:

lCN – номер логического канала;

synchFlag – номер новой динамической полезной нагрузки RTP;

ID_A – **generalID** источника;

IV – исходное значение/вектор для шифрования сеансового ключа;

IVs – исходное значение/вектор для шифрования ключа с "привязками" (salting key);

ENC_MK,IV,sc(K) – обозначает шифрование нешифрованного текста *K* с использованием ключа *M*, исходного вектора *IV* [и ключа с "привязками" *sc*, только при EOFB];

KS – ключ с "привязками" (salting key) для шифрования мультимедийной информации (только в режиме EOFB);

K – сеансовый ключ для нешифрованного текста;

sc – нешифрованный ключ с "привязками" (salting key) при использовании для шифрования сеансового ключа режима EOFB;

ksc – нешифрованный ключ с "привязками" (salting key) при использовании для шифрования ключа с "привязками" (salting key) режима EOFB;

s2M/m2S – флаг **direction** (H.235 только версия 3) (s2m = ведомый–ведущий, m2s = ведущий–ведомый);

[] – обозначает необязательную часть.

Методы обновления ключей, описываемые в следующих разделах, могут задействовать режим шифрования EOFB для защиты данных передаваемых ключей. Для того, чтобы задействовать режим EOFB для защиты данных ключей таким же образом, как и для защиты полезной мультимедийной нагрузки, необходимо использовать дополнительный ключ с "привязками" (sc или ksc).

B.2.6.2 Усовершенствованное обновление ключей

Конечные точки H.235 версии 3 и последующих версий должны будут выполнять процедуру обновления явных/неявных квитированных ключей. Цель этого – обеспечить надежные методы обновления ключей на основе метода обновления неквитированных ключей, представленного в версиях H.235, предшествующих версии 3. Возможности такой процедуры должны быть согласованы, используя индикацию возможностей версии 3, согласно B.2.3.

На рисунке B.1.2 представлены процедуры обновления ключей для логического канала, владельцем которого является ведомый объект. В случае, если ведомый объект инициирует обновление ключей и запрашивает новый сеансовый ключ у ведущего объекта, то ведомый объект должен послать **MiscellaneousCommand** ведущему объекту, в которой элемент **logicalChannelNumber** должен будет содержать номер логического канала (определяемый ведомым объектом), элемент **sharedSecret** должен быть установлен в истинное значение, флаг **direction** должен быть установлен в **slaveToMaster**, а номер новой динамической полезной нагрузки должен быть запрошен в элементе **synchFlag** в рамках **EncryptionUpdateRequest**. В ином случае, если ведущий объект инициирует обновление ключей, то нет необходимости посылать это сообщение **EncryptionUpdateRequest**.

Ведущий объект, то ли в ответ на запрос ведомого объекта, то ли сам по себе, должен будет выдать команду **EncryptionUpdateCommand**, в которой элемент **logicalChannelNumber** должен будет содержать номер логического канала, флаг **direction** должен быть установлен в **slaveToMaster** в рамках **MiscellaneousCommand**, а элемент **synchFlag** в рамках **encryptionSync** должен будет отражать номер новой динамической полезной нагрузки. Элемент **h235key** должен будет переносить новый сеансовый ключ. Этот элемент **h235key** должен будет содержать идентификатор ведущего объекта в **generalID** и используемый исходный вектор *IV* в **paramS**. Шифрованный сеансовый мультимедийный ключ должен будет передаваться в рамках **encryptedSessionKey**, при этом функция шифрования должна будет применяться к сеансовому ключу ведущего объекта, а исходное значение в **paramS** – к сеансовому ключу *K*. При режиме EOFB нешифрованный ключ с "привязками" (salting key) передается в **ClearSalt** в рамках **paramS** (*sc*). Элемент **encryptedSaltingKey** должен будет переносить шифрованный мультимедийный ключ с "привязками" (salting key), причем функция шифрования должна будет применить сеансовый ключ ведущего объекта и исходное значение **paramSsaltIV** к мультимедийному ключу с "привязками" (salting key) *KS*. При режиме EOFB нешифрованный ключ с "привязками" (salting key) (*ksc*) передается в **ClearSalt** в рамках **paramSsalt**. Элемент **clearSaltingKey** может содержать нешифрованный мультимедийный ключ с "привязками" (salting key), причем **encryptedSaltingKey** должен оставаться незаполненным, и наоборот. Добиться передачи нешифрованного ключа с "привязками" (salting key) можно только в том случае, если от этого не пострадает безопасность данных, в любом другом случае рекомендуется, чтобы мультимедийный ключ с "привязками" (salting key) шифровался.

Ведущий объект должен быть готов к приему зашифрованной мультимедийной информации под защитой нового сеансового ключа по представлении ему команды **EncryptionUpdateCommand**, но до получения **EncryptionUpdateAck** должен продолжать использовать старый сеансовый ключ. Ведущий объект может использовать новый сеансовый ключ, начиная с получения **encryptionUpdateAck**, тогда как ведомый объект может использовать новый сеансовый ключ, начиная с получения **EncryptionUpdateCommand**.

ПРИМЕЧАНИЕ 1. – Ведущий объект может выбрать любое значение типа полезной динамической нагрузки для ведомого объекта, так как тип полезной нагрузки связан именно с этим портом мультимедийного канала.

ПРИМЕЧАНИЕ 2. – Нет никакой необходимости для ведомого объекта в явно подтвержденном получении нового ключа. Ведущий объект в состоянии проследить получение сформированного ведомым объектом ключа при получении мультимедийной информации, шифрованной в соответствии с новым типом полезной нагрузки.

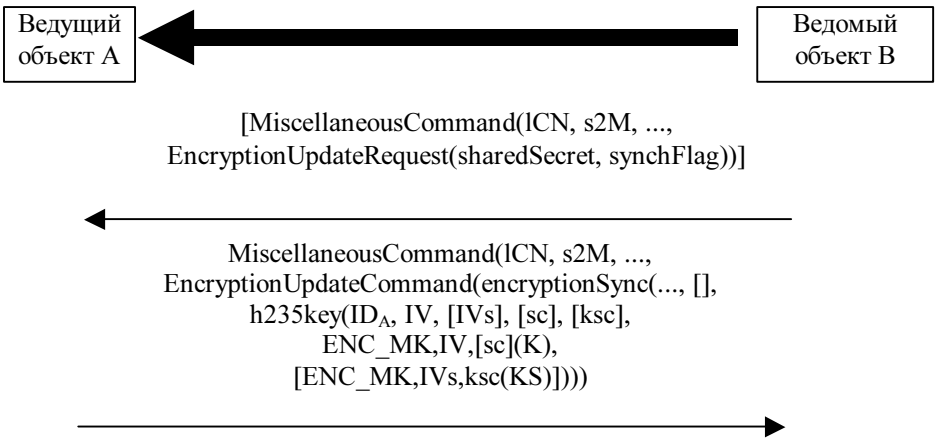


Рисунок B.1.2/H.235 – Обновление сеансового ключа в логическом канале ведомого объекта

На рисунке B.1.3 представлены процедуры обновления ключа для логического канала, владельцем которого является ведущий объект. В случае, если ведомый объект инициирует обновление ключа и запрашивает новый сеансовый ключ от ведущего объекта, ведомый объект должен послать команду **MiscellaneousCommand** к ведущему объекту, причем **logicalChannelNumber** должен содержать номер логического канала, элемент **sharedSecret** должен быть установлен в истинное значение, флаг **direction** должен быть установлен в **masterToSlave**. Если, в ином случае, ведущий объект инициирует обновление ключа, то это сообщение **EncryptionUpdateRequest** посылать не надо.

Ведущий объект, то ли в процессе ответа на запрос ведомого объекта, то ли сам по себе, должен будет выдать команду **EncryptionUpdateCommand**, причем элемент **logicalChannelNumber** должен будет содержать номер логического канала, флаг **direction** должен быть установлен в **masterToSlave**, элемент **encryptionSync** должен предоставить **synchFlag** с номером новой динамической полезной нагрузки. Элемент **h235key** должен будет переносить новый сеансовый ключ. **h235key** должен содержать идентификатор ведущего объекта в **generalID** и используемый исходный вектор *IV* в **paramS**. Шифрованный мультимедийный сеансовый ключ должен быть передан в рамках **encryptedSessionKey**, причем функция шифрования должна будет применить ключ ведущего объекта и это исходное значение в **paramS** к сеансовому ключу *K*. При режиме EOFB нешифрованный ключ с "привязками" (salting key) передается в **ClearSalt** в рамках **paramS** (*sc*). При режиме EOFB, элемент **encryptedSaltingKey** должен будет передавать шифрованный мультимедийный ключ с "привязками" (salting key), причем функция шифрования должна будет применить сеансовый ключ ведущего объекта и исходное значение **paramSsaltIV** к ключу с "привязками" (salting key) *KS*. При режиме EOFB нешифрованный ключ с "привязками" (salting key) (*ksc*) передается в **ClearSalt** в рамках **paramSsalt**. Элемент **clearSaltingKey** может содержать нешифрованный мультимедийный ключ с "привязками", в этом случае **encryptedSaltingKey** должен оставаться незаполненным, и наоборот. Добиться передачи нешифрованного ключа с "привязками" (salting key) можно только в том случае, если от этого не пострадает безопасность данных, в любом другом случае рекомендуется, чтобы мультимедийный ключ с "привязками" (salting key) шифровался.

Ведомый объект должен будет подтвердить получение нового сеансового ключа путем ответа посредством команды **MiscellaneousCommand**, причем элемент **logicalChannelNumber** должен будет содержать номер логического канала, а **encryptionUpdateAck** должен будет отражать номер новой динамической полезной нагрузки в **synchFlag**.

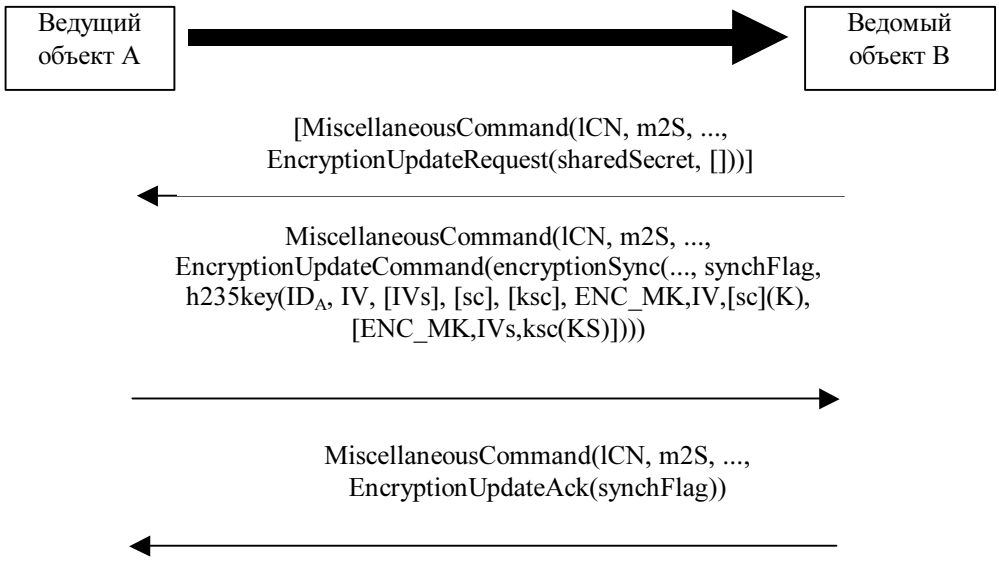


Рисунок B.1.3/H.235 – Обновление сеансового ключа в логическом канале ведущего объекта

B.2.6.3 Обновление ключа, базирующегося на типе полезной нагрузки, и синхронизация

Исходный ключ шифрования предоставляется ведущим объектом, наряду с номером динамической полезной нагрузки, в **synchFlag** (посредством **EncryptionSync** в Рек. МСЭ-Т H.245). Получатель(и) потока мультимедийной информации должен будет приступить к начальному применению этого ключа по получении этого номера полезной нагрузки в заголовке RTP.

Если обусловленный логический канал переносит только один тип полезной нагрузки, то тогда величина **synchFlag** может заменить тип согласованной полезной нагрузки в заголовке RTP. Если, с другой стороны, обусловленный логический канал может переносить несколько типов полезной нагрузки (даже, если только в отдельных пакетах RTP), то тогда пакеты RTP должны форматироваться, как описано в Стандарте RFC 2198, посредством значения **synchFlag**, действующего в качестве полезной нагрузки инкапсулированного типа, и фактического типа(ов) полезной нагрузки, размещающихся в дополнительном блоке(ах) заголовка, как определено Стандартом RFC 2198.

Новый ключ(и) может быть распределен в любое время посредством ведущей конечной точки. Синхронизация нового ключа с потоком мультимедийной информации будет указываться путем изменения типа полезной нагрузки на новое динамическое значение.

ПРИМЕЧАНИЕ. – Конкретные значения не учитываются, так как они изменяются при каждом новом распределяемом ключе.

B.3 Вопросы, касающиеся протоколов RTP/RTCP

Использование шифрования потока RTP должно осуществляться в соответствии с общей методикой, рекомендованной в документе, указанном в [RTP]. Шифрование мультимедийной информации должно осуществляться попакетно, отдельно для каждого пакета[1]. Заголовок RTP не должен шифроваться. Для аудио/видео кодеков вся полезная нагрузка аудио/видео кодека, включая любой заголовок(ки) аудио/видео полезной нагрузки, должна быть зашифрована. Синхронизация новых ключей и зашифрованного текста основывается на динамическом типе полезной нагрузки (см. пункт B.2.6.3).

Предполагается, что шифрование применяется только к полезной нагрузке в каждом пакете RTP, причем заголовки RTP остаются в незашифрованном виде. Предполагается, что все пакеты RTP должны быть кратными целому числу октетов. Для настоящей Рекомендации не имеет значения, как именно пакеты RTP инкапсулируются на транспортном или сетевом уровнях. Необходимо, чтобы во всех режимах потерянные (или внеочередные) пакеты, помимо пакетов-заполнителей, дополнялись до соответствующего кратного числа октетов.

---

[1] Следует отметить, что если размер пакета RTP больше размера пакета MTU, то частичная потеря (фрагмента) приведет к тому, что весь пакет RTP будет невозможно расшифровать.

Необходимо, чтобы отсутствовал статус расшифровки потока, поскольку пакеты могут теряться; каждый пакет должен расшифровываться отдельно. К режиму действия алгоритмов формирования блоков должны предъявляться два следующих требования:

B.3.1 Векторы инициализации

Большинство режимов формирования блоков включают некое "образование цепочки"; каждый цикл шифрования некоторым образом зависит от выходных данных предыдущего цикла. Следовательно, в начале пакета для запуска процесса шифрования необходимо обеспечить некоторое начальное значение блока [обычно называемое Вектором Инициализации (IV)]. Независимо от того, сколько октетов потока обрабатывается в каждом цикле шифрования, длина IV всегда равна длине блока. Все режимы, за исключением режима Электронная Кодовая Книга (ECB), требуют вектор IV.

B.3.1.1 Вектор инициализации CBC

Вектор Инициализации (IV) необходим при использовании блочного шифра в режиме CBC для шифрования пакетированной полезной нагрузки RTP. Размеры IV такие же, как и размеры блока при конкретном блочном шифре. К примеру, размеры IV для DES и 3-DES составляют 64 бита, а для AES – 128 битов.

В случае CBC вектор IV должен быть структурирован из первых B (где B – это размер блока) октетов Seq#, сцепленных с Timestamp. Это образует структуру *SSTTTT*, где *SS* – 2-октетная Seq# RTP, а *TTTT* – это 4-октетная временная метка. Эта структура должна повторяться до тех пор, пока будут сформированы октеты B, усеченные как необходимо. К примеру, IV размером в 64 и 128 битов должны будут содержать *SSTTTTSS* и *SSTTTTSSTTTTSSTT*, соответственно. Необходимо отметить, что сформированный таким образом IV может создавать такую структуру ключа, которая считается "неустойчивой" для конкретного алгоритма.

B.3.1.2 Вектор Инициализации EOFB

Однозначный исходный вектор IV для каждого пакета RTP в режиме EOFB должен вычисляться следующим образом:

Каждый пакет RTP ассоциируется с явным 48-битовым индексом пакета i, как определено в [SRTP], где $i = 2^{16} * \text{ROC} + \text{SEQ}$, где SEQ – это порядковый номер, взятый из заголовка RTP, а ROC – это 32-битовый счетчик циклических "прокруток", подсчитывающий частоту конвертаций порядкового номера SEQ в 65535.

Первоначально счетчик циклических "прокруток" ROC должен быть установлен в 0. Каждый раз когда SEQ конвертируется по модулю 2^{16}, отправитель должен инкрементировать ROC по одному модулю 2^{32}.

Исходный вектор IV вычисляется как ($i \parallel T$ [$\parallel i \parallel T \parallel \ldots$]) при 48-битовом индексе i и 32-битовой временной метке T, взятой из заголовка RTP, сцепленными несколько раз до полного заполнения размеров блока. Символ || обозначает сцепление.

ПРИМЕЧАНИЕ. – Счетчик циклических "прокруток" и IV поддерживаются и вычисляются локально на каждой одноранговой стороне и не передаются.

Получатель, столкнувшись с потерянным или, повторно упорядоченным пакетом, должен вычислить и установить индекс i следующим образом:

$i = 2^{16} * v + \text{SEQ}$, где v выбирается из последовательности {ROC-1, ROC, ROC+1} по модулю 2^{32} таким образом, что v является величиной, наиболее приближенной (т. е. 2^{48}) к величине $2^{16} * \text{ROC} + s_l$, где s_l – это поддерживаемый порядковый номер на стороне получателя. После обработки пакета с использованием определенного индекса, получатель должен решить, надо ли обновлять s_l и ROC. К примеру, простой (но не устойчивый к ошибкам) метод состоит в простой установке set s_l в SEQ, (если $\text{SEQ} > s_l$), и, если использовалось значение $v = \text{ROC} + 1$, в обновлении ROC в v; за дальнейшей информацией обратитесь также к [SRTP, раздел 3.2.1].

B.3.2 Заполнение

В режимах ECB и CBC обработка входного потока происходит поблочно, и при том, что CFB и OFB могут обрабатывать входной сигнал с любым количеством октетов, N ($\leq B$), рекомендуется, чтобы $N = B$.

Существует два метода для обработки пакетов, полезная нагрузка которых не представляет собой множество блоков:

1) Принудительный "захват" (Ciphertext Stealing) неполных блоков зашифрованного текста для ECB и CBC; отсутствие какого-либо заполнения в режимах CFB и EOFB.

2) Заполнение таким образом, как предписывает [RTP, раздел 5.1].

[RTP, раздел 5.1] описывает метод заполнения, согласно которому полезная нагрузка должна добавляться ко множеству блоков. Последний октет должен быть установлен в соответствии с числом октетов-заполнителей (включая последний) и установкой бита *P* в заголовке RTP. Величина заполнения должна определяться посредством обычного согласования алгоритма шифрования.

Все реализации Рекомендации H.235 должны поддерживать оба метода. Действие используемого метода может быть затем прослежено следующим образом: если бит *P* установлен в заголовке RTP, то пакет вставляется, если же пакет не представляет собой множество *B* и бит *P* не установлен, тогда применяется метод Ciphertext Stealing, если пакет представляет собой множество *B*, а заполнение не применяется.

B.3.3 Защита RTCP

Применение криптографических методов к элементам RTCP подлежит дальнейшему изучению.

B.3.4 Защищенный поток полезной нагрузки

Применение сетей, основывающихся на H.323, к примеру, сети передачи с помощью модема поверх IP используют сигнализацию H.245 для создания и согласования речевого канала передачи данных и RTP для пакетизации Группового Потока Полезной Нагрузки (MPS).

При одиночном мультимедийном потоке с одним типом полезной нагрузки или с прямым исправлением ошибок корректирующими кодами (FEC) для другого канала, тип динамической полезной нагрузки в элементе **encryptionSync** должен будет заменить тип полезной нагрузки по умолчанию.

При инкапсулированных потоках (то есть, с избыточным кодированием или FEC для кодированных согласно Стандарту (RFC) 2198) тип динамической нагрузки в рамках **encryptionSync** должен будет заменить инкапсулированный тип полезной нагрузки.

При групповых потоках полезной нагрузки тип динамической полезной нагрузки в **syncFlag** элемента **encryptionSync** должен игнорироваться, а вместо него должны использоваться (необязательные) типы полезной нагрузки в рамках **multiplePayloadStreamElement**.

EncryptionUpdateCommand должна будет использоваться при процедуре обновления усовершенствованного ключа для распространения данных нового сеансового ключа (см. B.2.6.2), **multiplePayloadStream** используется только тогда, когда множественный поток полезной нагрузки должен быть перенастроен по ключу, в этом случае тип динамической полезной нагрузки в рамках **EncryptionSync** должен будет игнорироваться.

B.3.5 Взаимодействие с Рекомендацией J.170

Подлежит дальнейшему изучению.

B.4 Передача сообщений (RAS)/процедуры для аутентификации

B.4.1 Введение

В настоящем Приложении прямо не будет представлен какой-либо тип обеспечения секретности сообщений между контроллерами доступа и конечными точками. Существует два типа аутентификации, которые можно применять. Первый тип – аутентификация на основе симметричного шифрования, не требующая предварительного контакта между конечной точкой и контроллером доступа. Второй тип аутентификации основывается на подписке и существует в двух видах – пароль или сертификат. Все эти типы аутентификации получены исходя из процедур, приведенных в пунктах 10.1, 10.3.2, 10.3.3 и 10.3.4. В данном Приложении унифицированные метки (EPA и EPB), показанные в вышеприведенных пунктах, представляют собой соответственно конечную точку и "контроллер доступа".

B.4.2 Аутентификация конечной точки – контроллера доступа (не на основе подписки)

Этот механизм может обеспечивать контроллер доступа криптографическим каналом связи с конкретной конечной точкой, которая заранее зарегистрировалась и совпадает с той, которая выдает последующие сообщения RAS. Следует отметить, что при этом может не осуществляться аутентификация контроллера доступа в направлении к конечной точке, если не включен дополнительный элемент подписи. Установление идентифицирующих взаимоотношений происходит при выдаче терминалом сообщения

GRQ в соответствии с пунктом 7.2.1/H.323. Обмен по алгоритму Диффи-Хеллмана должен происходить вместе с выдачей сообщений **GRQ** и **GCF**, как показано в первой части пункта 10.1. Затем общий секретный "ключ" должен будет использоваться для любого дальнейшего запроса **RRQ**/**URQ**, передаваемого от терминала к контроллеру доступа. Если контроллер доступа работает в этом режиме и получает сообщение **GRQ** без маркера, содержащего *DHset* или приемлемое значение алгоритма, то он должен будет возвратить сообщение с кодом причины **securityDenial**, или с другим соответствующим секретным кодом ошибки, согласно B.2.2 в **DRJ**.

Общий секретный "ключ" алгоритма Диффи-Хеллмана, созданный во время обмена **GRQ**/**GCF**, может применяться для аутентификации последующих сообщений **xRQ**. Приведенные ниже процедуры должны использоваться для проведения аутентификации в этом режиме.

Терминал (**xRQ**):

1) Этот терминал должен передавать всю информацию в этом сообщении согласно соответствующим пунктам Рек. МСЭ-Т H.225.0.

2) Этот терминал должен будет шифровать **GatekeeperIdentifier** (возвращенный в **GCF**) с помощью общего секретного "ключа", который был согласован. Он должен передаваться в **clearToken** (см. п. 10.2) как **generalID**.

Должна будет производиться логическая операция "исключающее ИЛИ" с 16 битами **random**, а затем с **requestSeqNum** и каждыми 16 битами **GatekeeperIdentifier**. Если **GatekeeperIdentifier** не заканчивается на кратном 16 граничном значении, то последние 8 битов **GatekeeperIdentifier** должны быть подвергнуты логической операции "исключающее ИЛИ" с наименее значащим октетом случайной величины, а затем с **requestSeqNum**. **GatekeeperIdentifier** должен быть зашифрован посредством алгоритма, выбранного в **GCF** (algorithmOID), с использованием всего общего "ключа".

Эта процедура проиллюстрирована на примере ниже:

RND16: 16-битовое значение Random Value

SQN16: 16-битовое значение requestSeqNum

BMPX: X-ный символ протокола пакетного режима (BMP) в GatekeeperIdentifier

BMP1' = (BMP1) XOR (RND16) XOR (SQN16)

BMP2' = (BMP2) XOR (RND16) XOR (SQN16)

BMP3' = (BMP3) XOR (RND16) XOR (SQN16)

BMP4' = (BMP4) XOR (RND16) XOR (SQN16)

BMP5' = (BMP5) XOR (RND16) XOR (SQN16)

:

:

BMPn' = (BMPn) XOR (RND16) XOR (SQN16)

Для того чтобы криптографически связать это и последующие сообщения с исходным зарегистрированным лицом (конечной точкой, выдавшей **RRQ**), должно быть использовано самое последнее значение **random** (это значение может быть более новым, чем то, которое было возвращено в **RCF** в результате более позднего сообщения **xCF**).

Контроллер доступа (**xCF**/**xRJ**):

1) Контроллер доступа должен будет шифровать свой **GatekeeperIdentifier** (после вышеописанной процедуры) посредством общего секретного "ключа", соответствующего псевдониму конечной точки, и сравнивать его со значением в **xRQ**.

2) Контроллер доступа должен будет возвращать **xRJ**, если эти два зашифрованных значения не соответствуют друг другу.

3) Если **GatekeeperIdentifier** соответствует, то контроллер доступа должен будет применить любую местную логику и ответить сообщениями **xCF** или **xRJ**.

4) Если Контроллер доступа передает **xCF**, то оно должно содержать присвоенный **EndpointIdentifier** и новое случайное значение в поле **random** в **clearToken**.

Графическое представление этого обмена показано во второй части рисунка 1. Контроллер доступа знает, какой общий секретный "ключ" использовать для расшифрования идентификатора контроллера доступа по псевдониму в этом сообщении.

B.4.3 Аутентификация конечной точки – контроллера доступа (на основании подписки)

Все сообщения RAS, отличные от GRQ/GCF, должны содержать маркеры аутентификации, которые требуют конкретный режим работы. Существует три различных версии, которые могут быть реализованы в зависимости от требований и условий:

1) аутентификация на основе пароля с симметричным шифрованием;

2) аутентификация на основе пароля с хешированием;

3) аутентификация на основе сертификата с подписями.

Во всех случаях маркер будет содержать информацию, описанную в следующих подпунктах. Если контроллер доступа работает в защищенном режиме и получает сообщение RAS без приемлемого значения маркера, то он должен возвратить код причины **securityDenial** или другой приемлемый секретный код ошибки, соответствующий B.2.2, в сообщении об отказе. Во всех случаях маркер возврата от контроллера доступа – необязателен; если он опущен, то производится только односторонняя аутентификация.

B.4.3.1 Пароль с симметричным шифрованием

Этап обнаружения контроллера доступа (GRQ, GCF и GRJ) может быть незащищен, как показано на рисунке B.2, или же он может быть защищен посредством **cryptoTokens**.

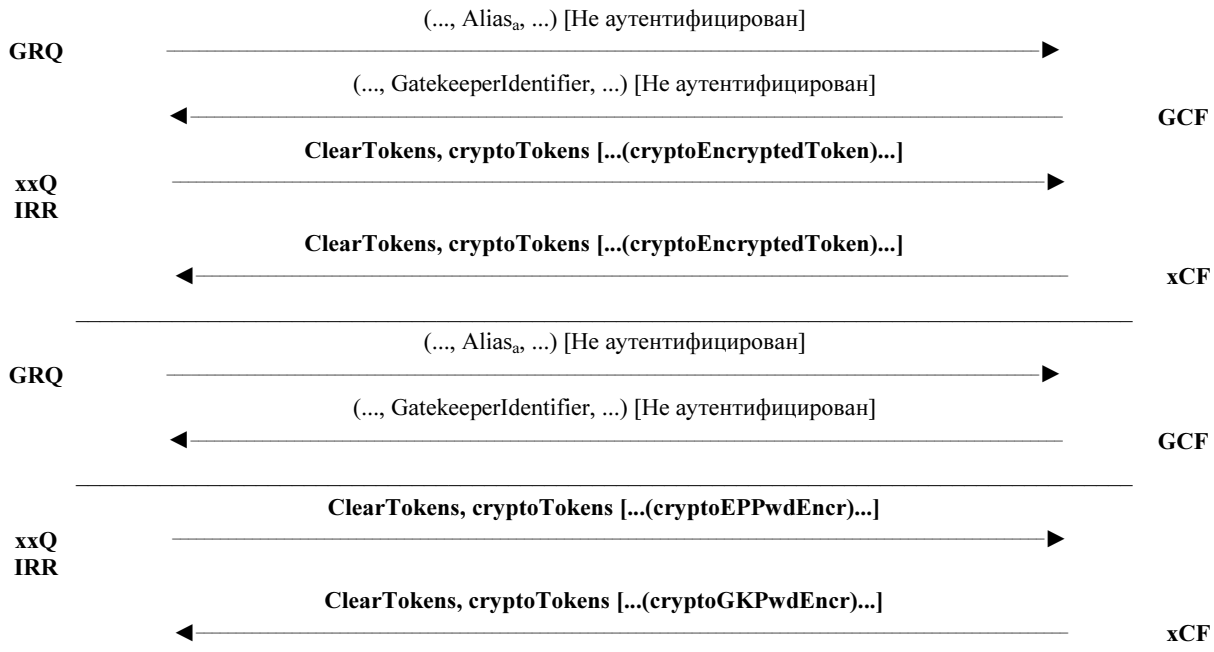


Рисунок B.2/H.235 – Пароль при симметричном шифровании

B.4.3.2 Пароль с хешированием

Этап обнаружения контроллера доступа (GRQ, GCF и GRJ) может быть незащищен, как показано на рисунке B.3, или же он может быть защищен согласно Приложению D посредством **cryptoTokens**.

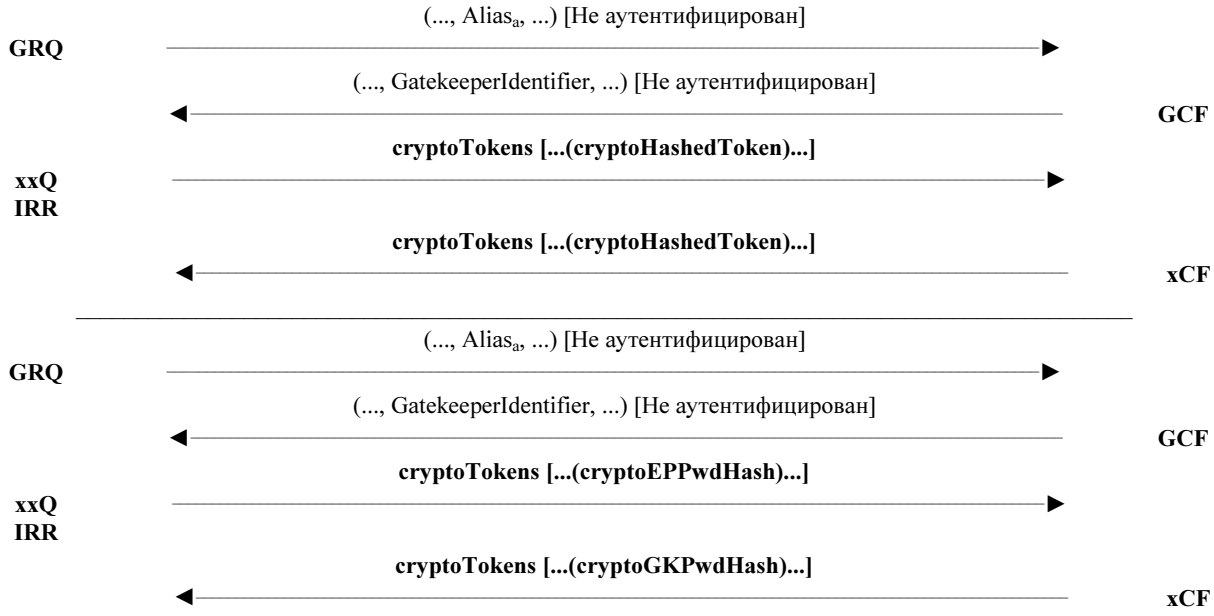


Рисунок B.3/H.235 – Пароль с хешированием

B.4.3.3 Аутентификация на основании сертификатов с подписями

Этап обнаружения контроллера доступа (GRQ, GCF и GRJ) может быть незащищен, как показано на рисунке B.4, или же он может быть защищен согласно Приложению E посредством **cryptoTokens**.

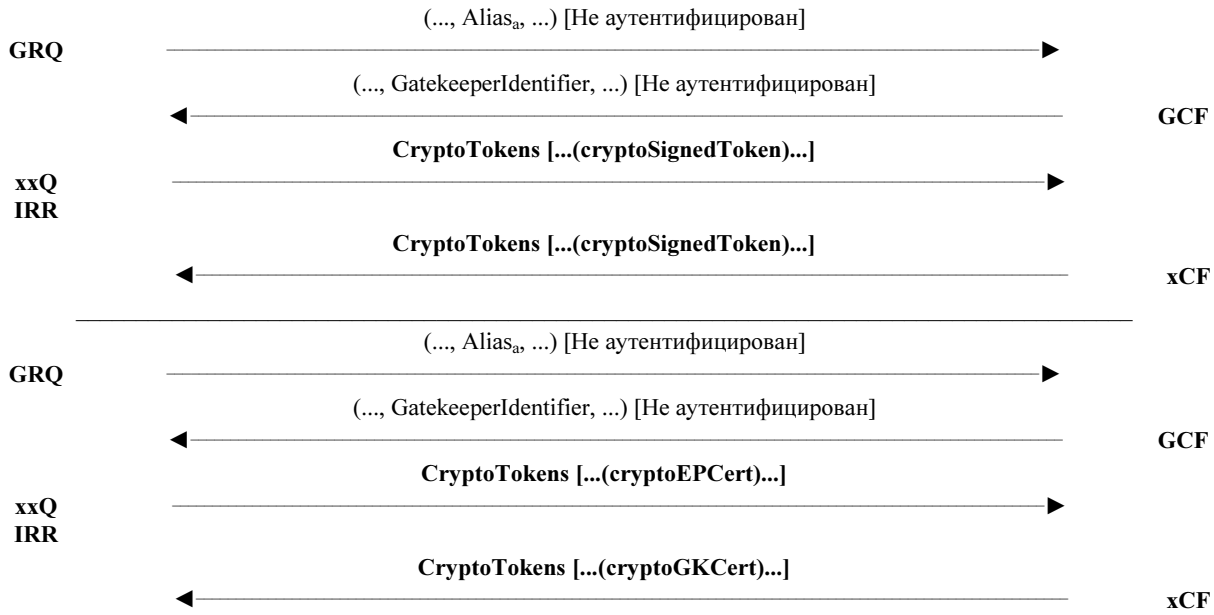


Рисунок B.4/H.235 – Аутентификация на основании сертификатов с подписями

B.5 Взаимодействие вне терминалов

B.5.1 Шлюз

Как указывалось в пункте 6.6, шлюз H.323 должен считаться доверительным элементом. Это распространяется и на шлюзы протоколов (H.323–H.320 и т. д.), и на шлюзы защиты (прокси/брандмауэры). Секретность мультимедийной информации может быть обеспечена между связывающимися друг с другом конечной точкой и шлюзовым устройством; но по умолчанию следует считать незащищенным процесс, который происходит в дальнем от шлюза конце.

B.6 Управление ключами в канале RAS

При некоторых ситуациях желательно распределять сеансовые ключи (RAS) от контроллера доступа к одной или нескольким конечным точкам под его контролем, или же от одной конечной точки к другой. Предложенный механизм подразумевает, что этот контроллер доступа и конечная точка имеют устойчивый общий секретный "ключ" или же знают открытый ключ друг друга. Одним из примеров такой ситуации может служить выдача маршрутизирующим контроллером доступа к конечной точке сеансового ключа в сообщении RAS, такого как **RCF** или **ACF**, для шифрования канала сигнализации, маршрутизируемого контроллером доступа. Другим примером может служить ситуация, при которой контроллер доступа выдает сеансовый ключ для использования его в шифровании, следующим за передачей RAS (к примеру, **RRQ** или **ARQ**).

Этот механизм подобен тому, что используется для распределения мультимедийных сеансовых ключей. Его можно использовать для того, чтобы избежать перегрузки при согласовании ключей в определенных ситуациях.

Для транспортировки ключей должно использоваться (H.235, версия 3) дополнительное поле **h235Key** в **ClearToken**. Гибкость элемента **H235Key** позволит осуществить транспортировку данных ключа шифрования, используя:

- защищенный канал (опция **secureChannel**), подразумевая, что RAS или канал сигнализации вызова имеют другие средства защиты (IPSEC/SSL и др.);
- общий "ключ" шифрования по незащищенному каналу (вариант **sharedSecret**), или что-то подобное, но предпочтительно – вариант **secureSharedSecret**;
- шифрование открытым ключом и сертификат по незащищенному каналу (опция **certProtectedKey**).

Вопросы использования сеансового ключа, которым обмениваются RAS, его применения к RAS, сообщений о посылке вызова и/или транспортных каналов подлежат дальнейшему изучению.

B.7 Псевдослучайная функция (PRF)

В данном пункте описывается псевдослучайная функция, предназначенная для получения динамических ключей исходя из данных статического ключа и случайной величины.

ПРИМЕЧАНИЕ. – Эта PRF идентична MIKEY PRF (см. [MIKEY]/RFC xxxx).

Метод формирования ключей имеет следующие входные параметры:

- *inkey*: ключ входа в функцию деривации.
- *inkey_len*: длина в битах ключа входа.
- *label*: специальная метка, зависящая от типа ключа, который нужно получить, и случайной величины **challenge**.
- *outkey_len*: требуемая длина в битах ключа выхода.

Псевдослучайная функция имеет следующий выход:

- *outkey*: выходной ключ требуемой длины.

Пусть HMAC (см. Стандарт 2104) будет базирующейся на сообщении SHA-1 [(см. ИСО/МЭК 10118-3)] функцией аутентификации. В соответствии со Стандартом 2246, определим:

$P\ (s, label, m) =$ HMAC ($s, A_1 \| label$) ||
HMAC ($s, A_2 \| label$) ||
HMAC ($s, A_m \| label$),

где:

A_0 = label,
A_i = HMAC (s, A_{i-1}).

При том, что SHA-1 ИСО/МЭК 10118-3 устанавливается по умолчанию, можно применять HMAC, используя другие функции хеширования; этот вопрос подлежит дальнейшему изучению.

Следующая процедура описывает псевдослучайную функцию, обозначаемую *PRF(inkey, label)*, применяемую для вычисления выходного ключа, выходной ключ:

- пусть $n = inkey\_len / 512$, округленное до ближайшего целого;
- разделите *inkey* на *n* блоков, $inkey = s_1 \| ... \| s_n$, где все s_i, за исключением, возможно, s_n, составляют 512 битов каждый;
- пусть $m = outkey\_len / 160$, округленное до ближайшего целого.

Тогда, выходной ключ, *outkey*, получается в виде наиболее значащих битов *outkey_len*:

$PRF(inkey, label) = P(s_1, label, m)\ XOR\ P(s_2, label, m)\ XOR\ ...\ XOR\ P(s_n, label, m).$

Приложение C

Вопросы, касающиеся Рекомендации H.324

Подлежат дальнейшему изучению.

Приложение D

Базовый профиль защиты

D.1 Введение

В настоящем Приложении описаны простые, базовые профили защиты. Указанные профили защиты основываются на положениях Рек. МСЭ-Т H.235 и на существующих профилях защиты, определенных Европейским институтом стандартов в области электросвязи (ETSI) и Международным консорциумом по мультимедийной телеконференционной связи (IMTC). Эти профили защиты позволяют выбрать необходимые характеристики защиты из богатого набора вариантов, предлагаемого Рек. МСЭ-Т H.235.

D.2 Термины, принятые в спецификациях

Необходимо сначала дать некоторые пояснения относительно терминов, используемых в настоящем Приложении.

В данном Приложении определяется **базовый профиль защиты**. Базовый профиль защиты обеспечивает основную защиту простыми средствами с помощью защитных криптографических методов на основе паролей. При необходимости, для достижения конфиденциальности при обмене речевыми сообщениями базовый профиль защиты может использовать **профиль защиты с шифрованием речевых сообщений**. В

Приложении E описан более сложный профиль защиты, в котором применяются цифровые подписи и преодолены ограничения, имеющиеся в базовом профиле защиты.
В настоящем Приложении используются поля H.235 для обеспечения таких сетевых средств защиты, как аутентификация/контроль целостности для сигнальных сообщений H.323. Различные идентификаторы объектов (см. D.11) определяют, какое сетевое средство защиты на самом деле выбрано и какая версия протокола настоящей Рекомендации в данный момент используется. Процедура I определяет способ реализации сетевых средств защиты посредством определенных механизмов защиты, таких как симметричные (хеширование по ключу) методы. Ссылки на идентификаторы объектов делаются посредством символов в тексте (например, "A"), см. также пункт 5.

При том, что сетевые средства защиты в виде контроля целостности сообщений всегда осуществляют и аутентификацию сообщений, обратное утверждение не всегда верно. На практике, объединенные сетевые средства защиты в виде аутентификации и контроля целостности используют одни и те же данные ключей, и это не ослабляет защиту.

Более того, вся последовательная информация о защите вводится в элемент **CryptoHashedToken**. Эта информация заново вычисляется при каждом переприеме.

Обычно, пароль, сеансовый ключ и общий "ключ" объединены тем, что все они применяются в симметричной криптографии между двумя (или более) объектами. Различие между паролем и сеансовым ключом/общим "ключом" состоит в способе реального применения ключей, например, пароли – для аутентификации и авторизации, сеансовые ключи – для шифрования. Термин общий "ключ" в некоторой степени нейтрален, поскольку он на самом деле не связан ни с каким конкретным применением.

Пароль (он может также рассматриваться как общий "ключ") используется для аутентификации/контроля целостности для сообщений RAS и H.225.0, поскольку этот элемент может вводиться пользователем. У пароля обычно более длительный срок службы; пароль известен *заранее* и может быть определен как часть всего процесса подписки пользователя. Некий алгоритм (например, конвейерная пересылка пароля посредством алгоритма хеширования) может преобразовывать этот пароль для более удобной обработки его в протоколах с тем, чтобы получить фиксированную длину.

С другой стороны, **сеансовый ключ** для шифрования потоков мультимедийной информации генерируется ведущим терминалом только для конкретного сеанса в RTP (по открытому логическому каналу (OLC)), максимум – для одного вызова. Сформированный сеансовый ключ зашифровывается с помощью ключа, полученного из согласованного **общего** "**ключа**" Диффи-Хеллмана, который рассчитали обе конечные точки. В этом случае общий "ключ", основанный на алгоритме Диффи-Хеллмана (DH), выступает как основой ключ для защиты сеансового ключа (ключей).

H.235 **ClearToken** имеет поле, называемое **random** и содержащее 32-битовую целую величину. Это поле используется следующим образом: **random** – это на самом деле равномерно возрастающее число, начинающееся с любого значения и увеличивающееся с каждым исходящим сообщением. Поле **random** применяется как дополнительная величина "рандомизации" для входа в функцию хеширования по ключу в том случае, если несколько сообщений выдаются сразу друг за другом, и при этом все же они передают идентичные временные метки. Это может произойти, когда часы Универсального скоординированного времени (UTC) не обеспечивают достаточного разрешения по времени. По сути, полученное хеш-значение или контрольный признак целостности выглядят по-разному из-за меняющейся величины **random**. Это нужно для противодействия повторным попыткам нарушения защиты. Для простоты реализации здесь отдается предпочтение увеличивающемуся счетчику, а не фактической случайной последовательности. Получатель может сохранять полученные пары **timestamp/random** в течение периода, определяемого окном местного времени[2]. Повторные попытки нарушения защиты могут быть выявлены, когда одна и та же пара **timestamp/random** возникает дважды.

Этот профиль предписывает, "установить **generalID** в **ClearToken** равным идентификатору получателя". На самом деле это означает, что для сообщений RAS, направляемых к контроллеру доступа, это – идентификатор GK, для сообщений RAS к конечной точке – это идентификатор конечной точки, для сообщений о посылке вызова H.225.0, направляемых к конечной точке, это – идентификатор вызываемой конечной точки, см. также пункт D.10**.**

sendersID должен быть установлен в значение идентификационной последовательности отправителя. Это фактически означает, что для сообщений RAS, направляемых к контроллеру доступа, это – идентификатор

---

[2] Окно времени компенсирует колебания синхронизированного времени и задержку транзита по сети.

конечной точки; для сообщений RAS, направляемых к конечной точке, это – идентификатор контроллера доступа, для сообщений о посылке вызова H 225.0, направляемых к контролеру доступа, это – идентификатор GK, а для сообщений о посылке вызова H.225.0, направляемых к конечной точке, это – идентификатор вызываемой конечной точки, см. также пункт D.10.

Блок – это базовый структурный элемент из пакетированных битов, который алгоритм блочного шифрования/дешифрования может зашифровать/расшифровать посредством элементарной криптографической операции; для стандарта DES и тройного DES размер блока составляет 64 бита, для AES размер блока составляет 128 битов.

Это приложение применимо для защиты целостности всего сообщения. При RAS H.225.0 защита целостности включает все сообщение RAS; при сообщении о посылке вызова, она охватывает все сообщение о посылке вызова H 225.0, включая заголовки Q.931.

Во избежание ссылок на торговую марку (RC2®) в настоящем Приложении делается ссылка на алгоритм шифрования, "совместимый с RC2".

В данной Рекомендации используются хорошо известные, касающиеся защиты термины такие, как ключ, управление ключами и SET, имеющие и другие значения в ином контексте (например, touch key pad – клавиатура с сенсорными клавишами, Q.931/Q.932 feature key management – управление функциональными ключами в Q.931/Q.932 и SET – Протокол защиты электронных транзакций).

D.3 Область рассмотрения

В настоящем Приложении описана простая защита для объектов H.323. Этот профиль защиты может применяться защищенными терминалами H.323, в том числе и **защищенным простым телефонным терминалом** (Защищенная конечная точка обмена аудиосигналами простого типа), описываемым в данном Приложении (см. D.6); такой профиль защиты может использоваться другими объектами H.323, например, шлюзами, контроллерами доступа, MCU.

D.4 Аббревиатуры

AES	Усовершенствованный алгоритм шифрования
BES	Внутренний сервер
CBC	Сцепление шифрованных блоков
DES	Стандарт шифрования блоков
DH	Алгоритм Диффи-Хеллмана
ECB	Электронная кодовая книга
EP	Конечная точка
ETSI	Европейский институт стандартов в области электросвязи
GK	Контроллер доступа
HMAC	Код аутентификации хешированных сообщений
IMTC	Международный консорциум по мультимедийной телеконференционной связи
IPSEC	Защита на уровне Интернет-протоколов
ITU	Международный союз электросвязи (МСЭ)
IV	Вектор инициализации
KS	Ключ с "привязками" (salting key) в режиме EOFB
MAC	Код аутентификации сообщений
MD5	Дайджест сообщений MD5
NAT	Трансляция сетевого адреса
OID	Идентификатор объекта
PFS	Полная секретность в прямом направлении
RAS	Протокол регистрации, допуска и статуса
RSA	Алгоритм шифрования открытым ключом (алгоритм Райвеста-Шамира-Адельмана)

RTP	Протокол передачи данных в режиме реального времени
SASET	Защищенная конечная точка обмена аудиосигналами простого типа
SET	Конечная точка простого типа
SHA	Алгоритм аутентификации и проверки целостности информации
TCP	Протокол управления передачей
TIPHON	Гармонизация телекоммуникационных и Интернет-протоколов в сетях
TLS	Протокол обеспечения безопасности транспортного уровня
VoIP	Передача речи поверх IP

D.5 Нормативные источники ссылок

Следующие Рекомендации МСЭ-Т и другие источники содержат положения, которые, будучи упомянутыми в качестве ссылок в данном тексте, составляют положения данной Рекомендации. На момент публикации указанные издания были действующими. Все Рекомендации и другие источники ссылок подлежат пересмотру; поэтому всем пользователям этих Рекомендаций предлагается рассмотреть возможность использования самого последнего издания этих Рекомендаций и других источников ссылок, перечисленных ниже. Перечень действующих рекомендаций МСЭ-Т публикуется регулярно. Ссылка на любой документ в рамках данной Рекомендации не придает ему, даже при том, что это отдельный документ, статуса Рекомендации.

AES [FIPS-197] Национальный институт стандартов, США, *"Усовершенствованный алгоритм шифрования (AES)"*, Федеральный стандарт на обработку информации (FIPS) Публикация 197, ноябрь 2001 г., http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf.

DES [FIPS-46-2] Национальный институт стандартов, США, *"Стандарт шифрования данных"*, Федеральный стандарт на обработку информации (FIPS). Публикация 46-2, декабрь 1993 г., http://www.itl.nist.gov/div897/pubs/fip46-2.htm.

DES [FIPS-74] Национальный институт стандартов, США, *"Руководство по реализации и применению стандарта шифрования данных"*, Федеральный стандарт на обработку информации (FIPS). Публикация 74, апрель 1981 г., http://www.itl.nist.gov/div897/pubs/fip74.htm.

DES [FIPS-81] Национальный институт стандартов, США, *"Режимы работы DES"*, Федеральный стандарт на обработку информации (FIPS). Публикация 81, декабрь 1980 г., http://www.itl.nist.gov/div897/pubs/fip81.htm.

[ИСО/МЭК 10118-3] ИСО/МЭК 10118-3:2004, *Информационные технологии – Методы защиты – Хеш-функции – Часть 3: Специализированные хеш-функции.*

[H.225.0] Рек. МСЭ-Т H.225.0, Версия 5 (2003 г.), *Протоколы передачи сигналов вызова и пакетирование потоков мультимедийной информации для мультимедийных систем связи в пакетном режиме.*

[H.235v1] Рек. МСЭ-Т H.235, Версия 1 (1998 г.), *Средства защиты и шифрования для мультимедийных терминалов серии H (терминалов H.323 и иных на основе H.245).*

[H.235v2] Рек. МСЭ-Т H.235, Версия 2 (2000 г.), *Средства защиты и шифрования для мультимедийных терминалов серии H (терминалов H.323 и иных на основе H.245).*

[H.245] Рек. МСЭ-Т H.245, Версия 10 (2003 г.), *Протокол управления для мультимедийной связи.*

[H.323] Рек. МСЭ-Т H.323 Версия 5 (2003 г.), *Мультимедийные системы связи, в пакетном режиме.*

[H.323 Приложение F] Рек. МСЭ-Т H.323 Приложение F (1999 г.), *Конечные точки простых типов.*

[RFC 2268] RFC 2268 (1998 г.), *Описание алгоритма шифрования RC2®.*

D.6 Базовый профиль защиты

В настоящем пункте описан базовый вариант простого профиля защиты.

D.6.1 Резюме

Базовый профиль защиты позволяет использовать модель с маршрутизацией посредством GK. Базовая защита применима в управляемой среде с симметричными ключами/паролями, присвоенными объектам (терминал–контроллер доступа, контроллер доступа–контроллер доступа, шлюз–контроллер доступа).

Данные профили обеспечивают следующие возможности:

– для сообщений RAS, H.225.0 и H.245:

  - Аутентификация пользователя для нужного объекта независимо от числа сетевых сегментов на прикладном уровне[3], которые проходит это сообщение.
  - Контроль целостности самого сигнального сообщения, включая критические части (поля) сообщений, поступающих на некоторый объект независимо от числа сетевых сегментов на прикладном уровне, которые проходит это сообщение.
  - Эти сетевые устройства защиты всего сообщения реализуются посредством последовательной (по сегментам) аутентификации сигнальных сообщений на прикладном уровне и контроля их целостности.

– для потока мультимедийной информации:

  - Конфиденциальность потока мультимедийной информации обеспечивается посредством симметричного шифрования.

Для предотвращения некоторых попыток нарушения защиты используется подходящий вариант вышеуказанных сетевых средств защиты. Сюда относятся:

- Попытки нарушения типа "отказ в обслуживании": Быстрая проверка криптографических хеш-значений может предотвратить такие попытки.
- Попытки нарушения защиты со стороны посторонних лиц: Последовательная (посегментная) аутентификация сообщения на прикладном уровне и контроль его целостности препятствуют подобным попыткам, если постороннее лицо, например, "недружественный" маршрутизатор, находится между сегментами прикладного уровня.
- Повторные попытки нарушения защиты: Использование временных меток и порядковых номеров предотвращает такие попытки.
- Спуфинг (имитация соединения): Аутентификация пользователя препятствует подобным попыткам.
- "Захват" соединения: Применение аутентификации/контроля целостности для каждого сигнального сообщения предотвращает такие попытки нарушения защиты.
- Для борьбы с подслушиванием потоков мультимедийной информации применяется шифрование и использование секретных ключей.

Другие особенности простого профиля защиты включают:

- Применение устойчивых, хорошо известных и широко используемых алгоритмов, основывающихся на данных IMTC/ETSI/IETF.
- Возможность поэтапного ввода устройств защиты, в соответствии с требованиями бизнес-модели.
- Возможность применения при различных сценариях реализации, как, например, в замкнутых группах, в условиях наращивания системы и при многосторонних конференциях.
- Профиль защиты, рассчитанный на только-аутентификацию, приемлем для обеспечения защиты при прохождении NAT/брандмауэра.

В таблице D.1 суммированы все определенные в настоящем Приложении процедуры, связанные с профилями защиты, соответствующими различным требованиям к защите. В эту таблицу включен

---

[3] Под сетевым сегментом здесь понимается доверительный элемент сети H.235 (например, GK, GW, MCU, прокси, брандмауэр). Таким образом, последовательная (от сегмента к сегменту) защита на прикладном уровне, при использовании симметричных методов шифрования, не обеспечивает действительной сквозной защиты между терминалами.

базовый профиль защиты (заштрихован вертикальными линиями, в электронном варианте – синий) и профиль защиты с шифрованием речевых сообщений (заштрихован горизонтальными линиями, в электронном варианте – зеленый).

Таблица D.1/H.235 – Резюме по профилям защиты, описанным в Приложении D

<table>
<tr><th rowspan="2">Сетевые средства защиты</th><th colspan="4">Функции вызова</th></tr>
<tr><th>RAS</th><th>H.225.0</th><th>H.245 (Примечание)</th><th>RTP</th></tr>
<tr><td>Аутентификация</td><td>Пароль HMAC-SHA1-96</td><td>Пароль HMAC-SHA1-96</td><td>Пароль HMAC-SHA1-96</td><td></td></tr>
<tr><td>Только-аутентификация</td><td>Пароль HMAC-SHA1-96</td><td>Пароль HMAC-SHA1-96</td><td>Пароль HMAC-SHA1-96</td><td></td></tr>
<tr><td>Защита от неподтверждения</td><td></td><td></td><td></td><td></td></tr>
<tr><td>Целостность</td><td>Пароль HMAC-SHA1-96</td><td>Пароль HMAC-SHA1-96</td><td>Пароль HMAC-SHA1-96</td><td></td></tr>
<tr><td></td><td></td><td></td><td></td><td>56-битовый DES | 56-битовый RC2-совместимый | 168-битовый тройной-DES | 128-битовый AES</td></tr>
<tr><td>Конфиденциальность</td><td></td><td></td><td></td><td>Режим CBC или EOFB</td></tr>
<tr><td>Управление доступом</td><td></td><td></td><td></td><td></td></tr>
<tr><td>Управление ключами</td><td>Присвоение паролей на основе подписки</td><td>Присвоение паролей на основе подписки | Обмен аутентифицированными ключами Диффи-Хеллмана</td><td>Управление интегрированными сеансовыми ключами H.235 (распределение ключей, обновление ключей, используя 56-битовый DES/56-битовый RC2-совместимый/168-битовый тройной-DES, 128-битовый AES)</td><td></td></tr>
<tr><td colspan="5">ПРИМЕЧАНИЕ. – Туннелированное H.245 или вложенные H.245 в рамках скоростного соединения H.225.0.</td></tr>
</table>

Для аутентификации пользователь должен будет применять схему, основывающуюся на пароле. Схема, основывающаяся на пароле, весьма рекомендуется для проведения аутентификации ввиду ее простоты и легкости реализации. Рекомендуемый подход к контролю целостности сообщений – это хеширование всех полей в сообщениях H.225.0 и сообщениях о посылке вызова (также при использовании схемы на основе пароля).

Защищенные объекты H.323 с этим профилем защиты осуществляют аутентификацию вместе с контролем целостности, применяя один и тот же общий механизм защиты.

Для реализации дополнительной конфиденциальности речевых сообщений предлагается схема шифрования с использованием алгоритмов AES-128, совместимых с RC2, DES или тройным DES на основе бизнес-модели и требований к экспорту. При некоторых условиях, которые уже обеспечивают некоторую степень конфиденциальности, шифрование речевых сообщений может не понадобиться. В этом случае согласование ключей по алгоритму Диффи-Хеллмана и иные процедуры управления ключами также не обязательны.

При внедрении профиля защиты с шифрованием речевых сообщений объекты H.323 должны будут применять 56-битовый стандарт DES в качестве алгоритма шифрования, выбираемого по умолчанию; они могут реализовывать 128-битовый AES или 168-битовый тройной DES, возможна также реализация экспортируемого шифрования с помощью алгоритма, совместимого с 56-битовым RC2.

Средства управления доступом четко не описаны; они могут определяться в рамках конкретной реализации после получения информации, передаваемой в сигнальных полях H.235 (ClearToken, CryptoToken).

В настоящей Рекомендации не описываются процедуры присвоения, а также контроля и административного управления паролем/"ключом" на основе подписки. Такие процедуры могут потребовать таких технических средств, которые выходят за рамки рассмотрения данного Приложения.
Объекты, участвующие в процессе связи, могут неявно определять использование базового профиля защиты или профиля защиты в виде подписи путем оценки передаваемых в сообщениях (**tokenOID** и **algorithmOID**; см. также пункт D.11) идентификаторов объектов защиты.

D.6.1.1 Базовый профиль защиты

Базовый профиль защиты применим в таких условиях, при которых предписанные пароли/симметричные ключи могут присваиваться защищенным объектам H.323 (терминалам) и сетевым элементам (GK, прокси). Он обеспечивает аутентификацию и контроль целостности или только-аутентификацию и передачу сигналов вызова для RAS H.225.0 и туннелируемых сообщений H.245 посредством хеширования HMAC-SHA1-96 на основе пароля, согласно Процедуре I. Установление соединения H.225.0 с помощью FastStart (от GK к GK или от терминала к терминалу) включает в себя интегрированное управление ключами по алгоритму Диффи-Хеллмана.

Вертикально заштрихованная область (синяя – в электронной версии) в таблице D.2 соответствует базовому профилю защиты.

Таблица D.2/H.235 – Базовый профиль защиты

Сетевые средства защиты	**Функции вызова**			
	RAS	**H.225.0**	**H.245**	**RTP**
Аутентификация и контроль целостности[4]	Пароль HMAC-SHA1-96	Пароль HMAC-SHA1-96	Пароль HMAC-SHA1-96	
Защита от неподтверждения				
Конфиденциальность				
Управление доступом				
Управление ключами	Присвоение паролей на основе подписки	Присвоение паролей на основе подписки		

Дополнительно, профиль защиты с шифрованием речевых сообщений может быть беспрепятственно объединен с базовым профилем защиты. Потоки аудиосигналов могут зашифровываться с помощью профиля защиты с шифрованием речевых сообщений на основе стандарта DES, алгоритма, совместимого с RC2, или тройного DES и с применением процедуры обмена аутентифицированными ключами Диффи-Хеллмана.

Базовый профиль защиты обеспечивает процедуру скоростного соединения с помощью элементов интегрированного управления ключами. Кроме того, предусмотрены средства сигнализации для туннелируемого обновления ключа H.245 и синхронизации. Для продолжительных вызовов эти сообщения требуют туннелирования сообщений H.245 в рамках H.225.0.

D.6.1.2 Профиль защиты с шифрованием речевых сообщений

Профиль защиты с шифрованием речевых сообщений, в отличие от базового профиля защиты, не является независимым профилем. Это скорее опция вышеуказанного профиля защиты, которая может использоваться совместно с ним. Этот профиль также основывается на определенных сетевых средствах защиты и является частью процедур передачи сигналов вызова и установления соединения, например, согласования ключей Диффи-Хеллмана и иных функций управления ключами.

Для достижения конфиденциальности речевых сообщений объекты H.323 могут использовать профиль с шифрованием речевых сообщений. Предлагаются три алгоритма шифрования: предложенные схемы представлют собой шифрование посредством стандарта AES, стандарта, совместимого с RC2, стандарта DES или тройного DES на основе бизнес-модели и требований к экспортируемости. В дополнение к режиму шифрования CBC, объекты H.323 могут реализовывать режим поточного шифрования EOFB. При некоторых условиях, когда уже обеспечена определенная степень

---

[4] Профиль защиты в виде только аутентификации не обладает свойством контроля целостности сообщений.

конфиденциальности, шифрование речевых сообщений может не понадобиться. В этом случае согласование ключей по алгоритму Диффи-Хеллмана и иные процедуры управления ключами также не обязательны.

При внедрении профиля защиты с шифрованием речевых сообщений объекты H.323 должны будут применять 56-битовый стандарт DES в качестве алгоритма, выбираемого по умолчанию; они могут реализовывать 128-битовый AES или 168-битовый тройной DES или же они могут реализовывать экспортируемое шифрование с помощью алгоритма, совместимого с 56-битовым RC2.

Профиль шифрования речевых сообщений определен в пункте D.2.

Таблица D.3/H.235 – Профиль шифрования речевых сообщений

<table>
<tr><th rowspan="2">Сетевые средства защиты</th><th colspan="4">Функции вызова</th></tr>
<tr><th>RAS</th><th>H.225.0</th><th>H.245</th><th>RTP</th></tr>
<tr><td>Аутентификация и целостность</td><td></td><td></td><td></td><td></td></tr>
<tr><td>Защита от неподтверждения</td><td></td><td></td><td></td><td></td></tr>
<tr><td></td><td></td><td></td><td></td><td>56-битовый DES
56-битовый совместимый с RC2
168-битовый тройной DES
128-битовый AES</td></tr>
<tr><td>Конфиденциальность</td><td></td><td></td><td></td><td>Режим CBC или EOFB</td></tr>
<tr><td>Управление доступом</td><td></td><td></td><td></td><td></td></tr>
<tr><td>Управление ключами</td><td></td><td>Обмен аутентифицированными ключами Диффи-Хеллмана</td><td>Управление интегрированными сеансовыми ключами H.235 (распределение ключей, обновление ключей)</td><td></td></tr>
</table>

D.6.2 Аутентификация и контроль целостности

В настоящем Приложении при описании сетевых средств защиты используются следующие термины:

- **Аутентификация и контроль целостности**: Это объединенный элемент сетевых средств защиты базового профиля, который поддерживает целостность сообщения наряду с аутентификацией пользователя. Пользователь может содействовать аутентификации, правильно применив процедуру манипуляции с общим секретным "ключом". Оба эти сетевые средства защиты имеют один и тот же механизм защиты.
- **Только-аутентификация**: Это сетевое средство защиты предлагается базовым профилем защиты в качестве опции, которая обеспечивает только-аутентификацию выбранных полей, но не обеспечивает целостность всего сообщения. Профиль защиты в виде только-аутентификации применим для сообщений сигнализации, проходящих NAT/брандмауэры. Пользователь может содействовать процессу аутентификации путем правильного применения процедуры манипуляции с общим секретным "ключом".

При использовании методов симметричных ключей аутентификация/контроль целостности сетевых средств защиты осуществляется только последовательно (от сегмента к сегменту).

D.6.3 Требования H.323

Предполагается, что объекты H.323, которые реализуют этот базовый профиль защиты, поддерживают следующие характеристики H.323:

- скоростное соединение;
- модель маршрутизации с помощью "контроллера доступа".

D.6.3.1 Описание процедуры

Ниже дано описание процедуры, используемой в этом профиле.

Процедура I представляет собой простой механизм аутентификации сигнальных сообщений на основе симметричных ключей, который базируется на применении общего пароля между двумя объектами (например, между контроллером доступа и конечной точкой H.323). Эта процедура обеспечивает аутентификацию и контроль целостности сообщений RAS, Q931 и H.245 (см. D.6.3.2).

Процедура IA представляет собой простой механизм только-аутентификации на основе симметричных ключей, который базируется на применении общего пароля между двумя объектами (к примеру, между

контроллером доступа и конечной точкой H.323). Эта процедура обеспечивает только-аутентификацию, но не обеспечивает целостность всего сообщения. Эта опция в виде только-аутентификации применима в тех сценариях, когда сигнальные сообщения H.323 проходят NAT/брандмауэры.

В зависимости от стратегии защиты, аутентификация может быть односторонней или взаимной с проведением аутентификации/контроля целостности также и в обратном направлении, что, таким образом, обеспечивает более высокий уровень защиты. Решение относительно проведения аутентификации/контроля целостности и в обратном направлении принимается контроллером доступа.

Контроллеры доступа, обнаружившие отказ при аутентификации и/или контроле целостности в сообщении о передаче сигналов вызова или сообщении RAS, которое было получено от защищенной конечной точки или от однорангового контроллера доступа, отвечают соответствующим сообщением отказа, указывающим на безуспешность защиты, посредством установки причины отказа в **securityDenial** или в другом коде ошибки, согласно B.2.2. В зависимости от способности к распознаванию попытки нарушения защиты и в качестве наиболее приемлемого способа реагирования на эту попытку, контроллер доступа, получая защищенный **xRQ** с неопределенными идентификаторами объектов (**tokenOID**, **algorithmOID)**, может ответить незащищенным **xRJ** и отвергнуть сообщение, установив причину отказа в **securityDenial**, или же он может сбросить это сообщение. Возникшее событие защиты должно быть зарегистрировано. С другой стороны, конечная точка должна будет удалить полученное незащищенное сообщение, сделать перерыв и повторно произвести попытку, выбирая различные OID. Подобным же образом, контроллер доступа, получая защищенное сообщение SETUP H.225.0 с неопределенными идентификаторами объектов (**tokenOID**, **algorithmOID)**, может ответить незащищенным сообщением RELEASE COMPLETE и отвергнуть сообщение, установив причину в **securityDenied**, или же он может сбросить это сообщение. Точно также, возникшее событие защиты должно быть зарегистрировано.

Неявная передача сигналов H.235 предусмотрена для указания на применение Процедуры I и соответствующего механизма защиты на основе значений идентификаторов объектов (см. также D.11) и заполненных полей сообщений.

Этот профиль не использует ICV-поля H.235; вместо этого контрольные признаки криптографической целостности обрабатываются как криптографические хеш-величины и помещаются в хеш-поля маркера **CryptoToken**.

D.6.3.2 Подробное описание процедуры аутентификации сигнальных сообщений на основе симметричных ключей (Процедура I)

Если используется Процедура I, то должны будут применяться следующие далее процедуры:

- 12-байтовая (96 битов) хеш-величина генерируется алгоритмом HMAC-SHA1-96 в качестве результирующего аутентификатора. Если ключ создается исходя из пароля, то для вычисления ключа исходя из пароля *должен* будет использоваться механизм, описанный в 10.3.5.

  ПРИМЕЧАНИЕ 1. – Когда секретный "ключ" выводится из введенного пользователем пароля, следует принять меры, гарантирующие его достаточную случайность. Рекомендуется, например, использовать действительно случайные "ключи" для получения секретного "ключа" или обеспечить достаточную длину случайных паролей.

- Поле **CryptoH323Token** в каждом сообщении RAS/H.225.0 должно будет содержать следующие поля:
  - **nestedCryptoToken**, содержащее **CryptoToken**, которое само содержит **cryptoHashedToken**, включающее следующие поля:
    - **tokenOID**, имеющее значение "A", которое указывает, что процесс вычисления ключа для аутентификации/контроля целостности учитывает все поля в сообщении RAS/H.225.0 и сообщении о посылке вызова.
    - **hashedVals**, содержащее поле **ClearToken**, используемое со следующими полями:
      - **tokenOID**, имеющее значение "T", которое указывает, что базовый **ClearToken** используется для аутентификации сообщений и повторной защиты, а также для управления ключами Диффи-Хеллмана, как описано в D.7.1. Вместо базового ClearToken, альтернативно, можно использовать другие ClearToken, с другими OID**.**
      - **timeStamp**, содержащее временную метку.
      - **random**, содержащее равномерно возрастающий порядковый номер. Этот номер позволяет создать два уникальных сообщения с одной и той же временной меткой (в пределах временного разрешения).
      - **generalID**, содержащее идентификатор получателя (только в случае одноадресных сообщений).

– **sendersID**, содержащее идентификатор отправителя.

– **dhkey**, используемое для передачи параметров Диффи-Хеллмана во время сообщений **Setup**-to-**Connect**, как это определяется настоящей Рекомендацией.

• **halfkey**, содержащее случайный открытый ключ одного участника.

• **modsize**, содержащее исходный ключ DH (см. таблицу D.4).

• **generator**, содержащее группу DH (см. таблицу D.4).

ПРИМЕЧАНИЕ 2. – Когда базовый профиль защиты используется без профиля защиты на основе шифрования речевых сообщений, не нужно пересылать никаких параметров Диффи-Хеллмана; поля **halfkey**, **modsize** и **generator** могут быть установлены в {"0"B, "0"B, "0"B}, а поле **dhkey** не должно присутствовать.

– **token**, содержащее **HASHED** с полями:

• **algorithmOID**, установленное в "U" и указывающее на использование HMAC-SHA1-96.

• **paramS**, установленное в NULL.

• **hash**, содержащее аутентификатор, вычисленный с использованием HMAC-SHA1-96. Этот аутентификатор может быть вычислен, исходя из:

– всех полей сообщения RAS/H.225.0 и сообщения о посылке вызова, если **tokenOID** в **CryptoHashedToken** установлен в "A" (указывающее на аутентификацию и контроль целостности).

tokenOID, установленный в "A", используется для обеспечения защиты туннелированных блоков H323-UU-PDU, включая все содержимое сообщений H.245; вычисление хеш-величины должно производиться по полному сигнальному сообщению о посылке вызова **H.225.0** со всеми полями, согласно процедуре, описанной в D.6.3.3.2.

• Аутентификатор проверяется в конце каждого оконечного участка канала (EP1-GK1, GK1-GK2, GK2-EP2, EP1-GK2, GK1-EP2 или EP1-EP2, в зависимости от конкретного случая), и повторно вычисляется перед отправкой сообщения на следующий участок.

ПРИМЕЧАНИЕ 3. – Аутентификатор вычисляется в расчете на каждое сообщение.

ПРИМЕЧАНИЕ 4. – Должен использоваться метод заполнения в рамках стандарта SHA1 (ИСО/МЭК 10118-3).

ПРИМЕЧАНИЕ 5. – Если используется комбинированная аутентификация и контроль целостности, то аутентификатор вычисляется для всего сообщения.

ПРИМЕЧАНИЕ 6. – Для того, чтобы предупредить возможность повторных попыток нарушения защиты настоятельно рекомендуется, чтобы реализации обеспечивали смену пароля (ключа) до полного обращения (или завершения цикла) равномерно увеличивающегося порядкового номера.

ПРИМЕЧАНИЕ 7. – Получатель способен обнаружить использование Процедуры I, путем определения **tokenOID** в рамках хешированного **EncodedGeneralToken** (обнаружив наличие "AB").

D.6.3.3 Вычисление хеш-величины на основе пароля

Как отправитель, так и получатель защищенного сообщения, аутентификации/контроля целостности, вычисляют хеш-величину по ключу, исходя из закодированных в ASN.1 полей сообщения (используя OID "A"). Для профиля "только-аутентификации" как отправитель, так и получатель, вычисляют хеш-величину по ключу, исходя из всего закодированного в ASN.1 Clear Token (используя OID "B").

D.6.3.3.1 HMAC-SHA1-96

HMAC-SHA1-96 является усеченной 96-битовой криптографической хеш-величиной 160-битового расчетного значения SHA1. В качестве результата должны будут использоваться 96 крайних левых битов представления этой хеш-величины в порядке поступления октетов по сети. RFC 2104 описывает процедуру с секретным "ключом" *K*, установленным в значение общего "ключа" (= значению SHA1-хешированного пароля), и *text*, помещенным в буфер сообщений.

D.6.3.3.2 Аутентификация и контроль целостности

Для аутентификации и контроля целостности сообщений (в случае, если применяется OID, установленный в "A") процедура выполняется следующим образом.

Отправитель сообщения должен будет вычислить хеш-величину следующим образом:

1) Установить хеш-величину равной конкретной заданной по умолчанию кодовой комбинации длиной в 96 битов. Точная комбинация битов не имеет значения, но хорошим вариантом является уникальная комбинация битов, которая не встречается в оставшемся сообщении.

2) Кодировать все сообщение в ASN.1; в случае RAS, должно быть включено полное сообщение RAS H.225.0; при сообщении о посылке вызова, должно быть включено полное сообщение о посылке вызова H.225.0.

3) Разместить[5] заданную по умолчанию кодовую комбинацию в закодированном сообщении; перезаписать всю определенную комбинацию битов с 96 нулевыми битами.

4) Вычислить криптографическую хеш-величину для закодированного в ASN.1 сообщения, используя HMAC-SHA1-96 (см. D.6.3.3.1).

5) Заменить заданную по умолчанию кодовую комбинацию в закодированном сообщении вычисленной хеш-величиной.

Получатель принимает сообщение и далее действует следующим образом:

1) Декодирует сообщение в ASN.1.

2) Извлекает полученную хеш-величину и сохраняет ее в локальной переменной RV.

3) Находит и локализирует хеш-величину RV в полученном закодированном сообщении.

ПРИМЕЧАНИЕ. – В тех редких случаях, когда подстрока хеш-величины может встретиться в полном сообщении несколько раз, этапы 3–6 должны выполнятся методом последовательных итераций, при этом отправные точки для поиска должны быть разными.

4) Перезаписывает всю комбинацию битов в закодированном сообщении с помощью 96 нулей.

5) Вычисляет криптографическую хеш-величину по закодированному сообщению, используя HMAC-SHA1-96 (см. D.6.3.3.1).

6) Сравнивает RV с вычисленной хеш-величиной. Сообщение считается ненарушенным только в том случае, если обе хеш-величины равны; в этом случае аутентификация прошла успешно, и процедура прекращается.

7) В противном случае, получатель повторяет этапы 3–7, восстанавливая RV на прежнем месте и производя поиск другого соответствия. Если ни одно сравнение соответствий не привело к получению правильной хеш-величины значения, то аутентификация была безуспешной, и сообщение было изменено (случайно или преднамеренно) во время пересылки.

D.6.3.3.3 Только-аутентификация (Процедура IA)

Терминалы могут отдать предпочтение реализации процедуры только-аутентификации (используя OID "B", см. E.18). В этом случае аутентификатор вычисляется по подмножеству (**ClearToken** внутри **CryptoToken**) сообщения RAS/H.225.0. Эта процедура только-аутентификации может быть пригодна для прохождения NAT/брандмауэров, которые изменяют IP-адреса/порты в пределах полезной нагрузки H.323.

Так как аутентификация охватывает только очень ограниченную часть сообщения, то процедура только-аутентификации, в отличие от Процедуры I, не обеспечивает целостность сообщения. Такая процедура только-аутентификации в меньшей степени обеспечивает защиту.

При процедуре только-аутентификации в защищенных сообщениях должны использоваться следующие поля:

- Поле **CryptoH323Token** в каждом сообщении RAS/H.225.0 должно содержать следующие поля:
  - **nestedCryptoToken**, содержащее **CryptoToken**, которое само содержит **cryptoHashedToken**, содержащее следующие поля:
    - **tokenOID**, установленное в:
      - "B" (см. E.18), указывающее, что вычисление при только-аутентификации включает все поля в **ClearToken**.

---

[5] Это может потребовать некоторых эмпирических шагов в том редком случае, когда заданная по умолчанию кодовая комбинация встречается в сообщении более одного раза.

- **hashedVals**, содержащее поле **ClearToken**, используемое со следующими полями:
  - **tokenOID**, установленное в:
    - "T" (как пример базового ClearToken для остального содержимого ClearToken) или любой, соответствующий любому предназначению подходящий для любых других целей OID.
  - **timeStamp** содержит временную метку;
  - **random** содержит равномерно увеличивающийся порядковый номер. Этот номер позволяет составить два сообщения с одной и той же временной меткой (в пределах разрешения по времени);
  - **generalID** содержит идентификатор получателя (только в случае одноадресных сообщений);
  - **sendersID** содержит идентификатор отправителя;
  - **dhkey**, используемое для передачи параметров Диффи-Хеллмана во время **Setup**-to-**Connect**, как описано в H.235.
    - **halfkey** содержит случайный открытый ключ одного участника.
    - **modsize** содержит исходный ключ DH (см. таблицу D.4).
    - **generator** содержит DH-группу (см. таблицу D.4).

    ПРИМЕЧАНИЕ 1. – Когда базовый профиль защиты используется без профиля защиты с шифрованием речевых сообщений, тогда никакие параметры Диффи-Хеллмана не должны пересылаться, а **dhkey** должен отсутствовать; **halfkey**, **modsize** и **generator** могут быть установлены в {'0'B,'0'B,'0'B}.
  - **token**, содержащее **HASHED** с полями:
    - **algorithmOID**, установленное в "U" и указывающее на применение HMAC-SHA1-96;
    - **paramS** установленое в NULL;
    - **hash**, содержащее аутентификатор, вычисленный с использованием HMAC-SHA1-96. Этот аутентификатор должен быть вычислен по:
      - всем полям **ClearToken**, если **tokenOID** в **CryptoHashedToken** установлен в "B" (указывая на процедуру только-аутентификации).
- Аутентификатор проверяется в конце оконечного участка каждого канала (EP1-GK1, GK1-GK2, GK2-EP2, EP1-GK2, GK1-EP2 или EP1-EP2, в зависимости от ситуации) и повторно вычисляется до отправления сообщения к следующему участку.

ПРИМЕЧАНИЕ 2. – Аутентификатор вычисляется прямо по **ClearToken**.

ПРИМЕЧАНИЕ 3. – Должен применяться метод заполнения в рамках стандарта SHA-1 [ИСО/МЭК 10118-3].

ПРИМЕЧАНИЕ 4. – Для того, чтобы предупредить возможность повторных попыток нарушения защиты настоятельно рекомендуется, чтобы реализации обеспечивали смену пароля (ключа) до полного обращения (или завершения цикла) равномерно увеличивающегося порядкового номера.

ПРИМЕЧАНИЕ 5. – Получатель способен обнаружить использование Процедуры IA, путем определения **OID** "B" в рамках **tokenOID**.

Аутентификатор должен вычисляться прямо по **ClearToken** внутри **CryptoH323Token** (то есть, **ClearToken**) маркера **token** в **cryptoHashedToken**. Криптографическая значение хеш-величина должна вычисляться по кодированной в ASN.1 битовой строке в **ClearToken**.

Конечные точки H.235, версии 1 и версии 2 могут использовать процедуру только-аутентификации, в этом случае надо использовать соответствующие OID для "B". Конечные точки H.235, версии 1 должны придерживаться процедуры, описанной в D.6.6.

D.6.3.4 Пример использования Процедуры I

На рисунках D.1–D.3 показано наличие общих "ключей" в конце каналов связи при различных комбинациях контроллера доступа и каналов прямой маршрутизации H.225.0. Независимо от модели вызова общий "ключ" всегда присутствует между EP и ее GK, чтобы обеспечить аутентификацию и контроль целостности сообщений RAS. Когда канал RAS и канал H.225.0 ограничены двумя одинаковыми узлами, для аутентификации и контроля целостности может использоваться один и тот же ключ как для сообщений RAS, так и для сообщений H.225.0.

На рисунке D.1 показан наиболее расширяемый сценарий, при этом обе конечные точки находятся в пределах зон, к которым применима модель маршрутизации с помощью GK. Все участвующие GK используют ключи совместно. Для обеспечения масштабируемости рекомендуется сценарий, изображенный на рисунке D.1.

ПРИМЕЧАНИЕ 1. – Этот сценарий не обеспечивает действительную сквозную защиту между конечными точками; вся система защиты зависит от доверительных промежуточных контроллеров доступа.

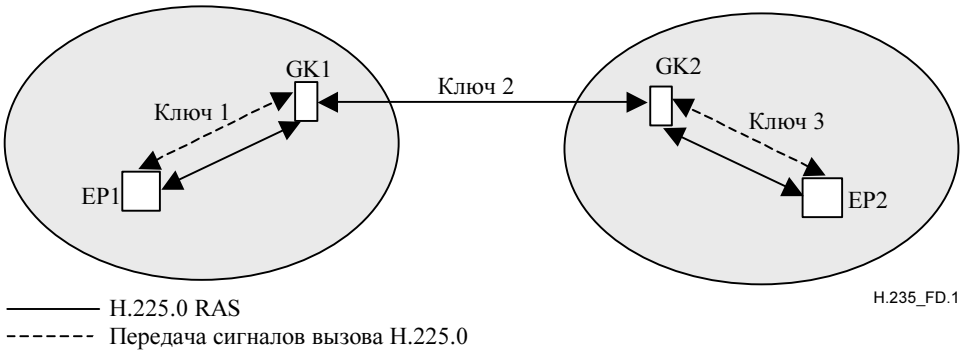


Рисунок D.1/H.235 – Пример использования Процедуры I в сценарии GK-GK, когда обе EP находятся в зонах маршрутизации с помощью GK

На рисунке D.2 показан смешанный сценарий, при этом одна EP располагается в пределах доступа, к которой применима модель маршрутизации с помощью GK, в то время как другая EP находится в зоне, к которой применима модель прямой маршрутизации. Этот сценарий может применяться в закрытых системах, где число EP2 и GK1 ограничено.

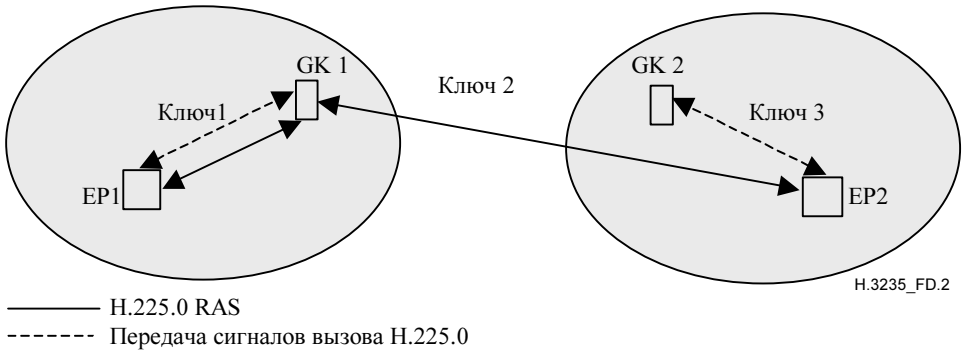


Рисунок D.2/H.235 – Пример использования Процедуры I в смешанном сценарии, когда EP1 находится в зоне маршрутизации с помощью GK, а EP 2 – в зоне прямой маршрутизации

На рисунке D.3 показан сценарий, при котором обе EP расположены в пределах зон, использующих модель прямой маршрутизации с помощью GK. Этот сценарий не имеет больших возможностей расширения, если участвуют много EP. В принципе, вместо этого рекомендуется использовать

Приложение E с Процедурами II/III. Для этого конкретного сценария и Процедур I, II или III также необходимы дополнительные меры защиты[6], которые не описаны в настоящей Рекомендации; этот вопрос подлежит дальнейшему изучению.

ПРИМЕЧАНИЕ 2. – Этот сценарий обеспечивает действительную сквозную защиту между конечными точками, вне зависимости от доверительных промежуточных контроллеров доступа.

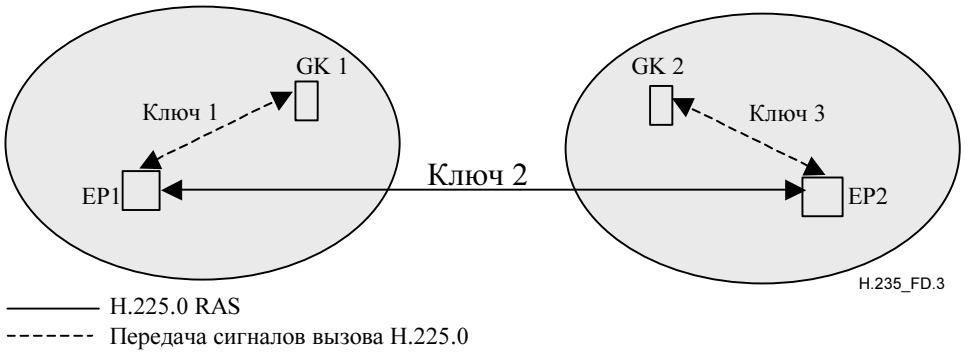


Рисунок D.3/H.235 – Пример использования Процедуры I в сценарии, когда обе EP находятся в зонах прямой маршрутизации с помощью GK

Рассмотрим вариант на рисунке D.1, при котором три пароля попарно разделены между EP1-GK1, между GK1-GK2 и между GK2-EP2. Три 20-байтовых ключа *"Ключ 1"*, *"Ключ 2"* и *"Ключ 3"* сформированы из этих паролей с помощью процедуры, описанной в 10.3.5. Для максимальной защиты рекомендуется сделать независимыми каждый из этих трех случайных паролей/ключей.

Ниже подробно показана процедура аутентификации и контроля целостности сообщений RAS, H.225.0 и H.245. Описываемый пример показывает конкретные параметры в модели маршрутизации с помощью GK; также возможны другие соответствующие и допустимые комбинации идентификаторов объектов в различных сценариях.

ПРИМЕЧАНИЕ 3. – Сценарии, показанные на рисунках 1–3, не допускают значительного расширения в том случае, если количество общих симметричных ключей (паролей) между GK (рисунок D.1), между GK и удаленными EP (рисунок D.2) или между EP (рисунок D.3) становится слишком большим.

D.6.3.4.1 Аутентификация и контроль целостности сообщений RAS

Рассмотрим случай, когда EP1 желает послать сообщение RAS, например, сообщение **ARQ** к GK1. EP1 генерирует временную метку и порядковый номер и включает их в поля **timeStamp** и **random**, соответственно, наряду с псевдонимом GK1 в **generalID** и идентификатором (ID) EP в поле **sendersID**. Эти поля присутствуют в поле **ClearToken** элемента **hashedVals**, присутствующего в **cryptoHashedToken** поля **CryptoToken** структуры **cryptoH323Token** сообщения **ARQ**.

Поле **tokenOID** в рамках **cryptoHashedToken** установлено в "A", которое указывает, что все поля в сообщении **ARQ** хешированы. **HASHED** в рамках **token** в **cryptoHashedToken** имеет **algorithmOID**, установленный в "U", что указывает на использование HMAC-SHAl-96, и **paramS**, установленный в NULL. Затем EP1 с помощью HMAC-SHA1-96 вычисляет аутентификатор, используя 20-байтовый ключ *"Ключ 1"*. Аутентификатор вычисляется по полному сообщению RAS.

EP1 включает вычисленный аутентификатор в рамках **hash** в поле **token** поля **cryptoHashedToken** маркера **CryptoToken**, присутствующего в **cryptoH323Token** сообщения **ARQ**. Затем сообщение **ARQ** посылается к GK1.

По получении сообщения **ARQ**, GK1 проверяет аутентификатор по нескольким критериям, которые включают:

---

[6] Обеспечение защиты от несанкционированных вызовов и неправильного употребления посредством авторизации вызова с помощью маркеров доступа, например, в шлюзах H.323.

- жизнеспособность **timeStamp**, уникальность **random**;
- тождество **generalID** и собственного идентификатора;
- соответствие аутентификатора в сообщении **ARQ** аутентификатору**,** вычисленному самим GK1.

D.6.3.4.2 Аутентификация и контроль целостности сообщений H.225.0

Рассмотрим случай, когда EP1 желает послать сообщение H.225.0, например, сообщение **Setup**, к EP2. EP1 генерирует временную метку и порядковый номер и включает их в поля **timeStamp** и **random**, соответственно, вместе с псевдонимом GK1 в **generalID** и идентификатором (ID) EP в поле **sendersID**. EP1 рассчитывает также полуключ Диффи-Хеллмана и включает параметры Диффи-Хеллмана **halfkey**, **modsize** и **generator** в поле **dhkey** в **ClearToken**. Эти поля присутствуют в поле **ClearToken** элемента **hashedVals**, присутствующего в **cryptoHashedToken** поля **CryptoToken** в **cryptoH323Token** сообщения **Setup**.

Поле **tokenOID** в **cryptoHashedToken** устанавливается в "A", указывая, что все поля в сообщении о посылке вызова H.225.0 хешированы. **HASHED** в рамках **token** в **cryptoHashedToken** имеет **algorithmOID**, установлен в "U", что указывает на использование HMAC-SHAl-96, и **paramS**, установленный в NULL. Затем, EP1 с помощью алгоритма HMAC-SHA1 вычисляет аутентификатор, используя 20-байтовый ключ *"Ключ 1"*. Аутентификатор вычисляется в соответствии с выбранным методом хеширования (A), учитывающим полное сообщение о посылке вызова H.225.0.

EP1 включает вычисленный аутентификатор в рамках **hash** в поле **token** поля **cryptoHashedToken** маркера **CryptoToken**, присутствующего в **cryptoH323Token** сообщения **Setup**. Затем сообщение **Setup** посылается к GK1.

По получении сообщения **Setup**, GK1 проверяет аутентификатор по нескольким критериям, которые включают:

- жизнеспособность **timeStamp**, уникальность **random**;
- тождество **generalID** и собственного идентификатора;
- проверку параметров Диффи-Хеллмана, например, проверку того, являются ли правильными 1024-битовый исходный ключ и поле **generator**. Проверка защищенности параметров Диффи-Хеллмана является трудоемким процессом, и она может производиться только тогда, когда этого требует локальная стратегия;
- соответствие аутентификатора в сообщение **Setup** аутентификатору, вычисленному самим GK1.

Если аутентификатор успешно проверен, то GK1 вычисляет новый аутентификатор, чтобы ввести (заменить) его в сообщении **Setup** перед последующей передачей его к GK2. GK1 заменяет поля **timeStamp**, **random**, **sendersID** и **generalID** полем **ClearToken** в **hashedVals**, используя значения, соответствующие участку маршрута GK1-GK2. Поле **timestamp** содержит текущую временную метку, поле **random** – следующий равномерно увеличивающийся порядковый номер для участка маршрута GK1-GK2, поле **generalID** – псевдоним GK2, а **sendersID** – псевдоним GK1. GK1 также включает полученные параметры Диффи-Хеллмана в поле **dhkey** в **ClearToken**.

Затем GK1 вычисляет новый аутентификатор для этого сообщения о посылке вызова H.225.0, используя ключ *"Ключ 2"* и алгоритм HMAC-SHA1-96 (**algorithmOID** = "U"), вставляет его в **hash** в рамках **token** и передает сообщение **Setup** к GK2.

По получении сообщения **Setup,** GK2 проверяет аутентификатор, вычисляет новый аутентификатор после изменения соответствующим образом полей **ClearToken** в **hashedVals**, вставляет его в поле **hash** и передает сообщение **Setup** к EP2.

D.6.3.4.3 Аутентификация и контроль целостности сообщений H.245

Рассмотрим случай, когда EP1 желает послать сообщение H.245, например, сообщение **TerminalCapabilitySet**, к EP2. EP1 проверяет необходимость посылки сообщения H.225.0 к GK1. Если эта необходимость существует, то сообщение H.245 туннелируется в рамках сообщения H.225.0. Поля в сообщении H.225.0 устанавливаются в соответствии с приведенным выше описанием передачи сообщений H.225.0. Поскольку сообщение H.245 туннелируется, то **h323-uu-pdu** в сообщении **h323-UserInformation** имеет свои поля, установленные следующим образом:

- Поле **h323-message-body** устанавливается в значение, которое соответствует типу передаваемого сообщения H.225.0.
- **h245Tunnelling** устанавливается в значение "TRUE".
- **h245Control** содержит цепочку октетов PDU H.245.

EP1 генерирует **CryptoToken** для сообщения H.225.0, устанавливает **tokenOID** в "A", что указывает на аутентификацию и контроль целостности, устанавливает в "T" полям **timeStamp**, **random**, **sendersID**, **generalID** и **tokenOID** в **ClearToken** в **hashedVals**, устанавливает **algorithmOID** в "U", что свидетельствует об использовании HMAC-SHA1-96, а **hash** устанавливает равным вычисленному аутентификатору хеш-величины по всем полям сообщения о посылке вызова H.225.0.

Однако, при отсутствии отложенной передачи сообщений H.225.0, сообщение H.245 туннелируется в рамках специального сообщения **facility** H.225.0. Поле **h323-uu-pdu** в сообщении **h323-UserInformation** устанавливает свои поля следующим образом:

- Поле **h323-message-body** устанавливается в **facility**, которое содержит:
  - **reason**, установленное в **undefinedReason**;
  - **tokens** и **cryptoTokens**, установленные, как для любого сообщения H.225.0.
- **h245Tunnelling**, устанавливается в "TRUE".
- **h245Control**, содержит цепочку октетов PDU H.245.

Как описано выше, EP1 генерирует **CryptoToken** как элемент сообщения **facility** H.225.0. Сообщение **facility** затем передается от EP1 к GK1.

В любом случае (независимо от того, отложена ли передача сообщения H.225.0 или используется специальное сообщение **facility** H.225.0), GK1 проверяет аутентификатор по получении этого сообщения. Затем, если передача сообщения H.225.0 задерживается на участке GK1-GK2, то сообщение H.245 туннелируется в рамках этого сообщения; или же, оно туннелируется внутри специального сообщения **facility** H.225.0. Как и в случае передачи любого сообщения H.225.0, вычисляется новый аутентификатор для сообщения H.225.0 до его передачи от GK1 до GK2. Процесс повторяется для участка GK2-EP2.

D.6.4 Сценарий прямой маршрутизации

Защищенные объекты H.323 могут обмениваться сообщениями не только в условиях маршрутизации с помощью GK, как отмечается в настоящей Рекомендации, но они могут также использовать модель прямой маршрутизации. Эта модель требует дополнительных мер защиты (маркеры доступа), которые не нужны в более простых условиях маршрутизации с помощью GK. Обеспечение защиты модели прямой маршрутизации является предметом дальнейшего изучения.

D.6.5 Поддержка внутренних серверов

Защищенные объекты H.323 могут использовать внутренние серверы согласно процедуре, описанной в 1.4.6.

D.6.6 Совместимость с протоколом H.235 версии 1

Несмотря на то, что эти профили защиты разработаны для протокола H.235 версии 2 (Рек. МСЭ-Т H.235 (2000 г.)), также возможно их применение для протокола H.235 версии 1 (Рек. МСЭ-Т H.235 (1998 г.)) с некоторыми небольшими модификациями. Получатель способен обнаружить присутствие версии протокола H.235 отправителя, определяя идентификаторы объектов профиля защиты (см. D. 11).

Реализации H.235, версии 1 (Рек. МСЭ-Т H.235 (1998 г.)):

- Не устанавливают и не определяют значение sendersID в ClearToken.
- Не могут использовать внутренние серверы, как это описано в D.6.5.

D.6.7 Режим многоадресной передачи

Многоадресные сообщения H.225.0 типа **GRQ** или **LRQ** не должны включать **CryptoToken** согласно Процедуре I. Если такие сообщения посылаются в один адрес, то это сообщение должно включать **CryptoToken**.

D.7 Профиль защиты на основе шифрования речевых сообщений

Общая процедура формирует общий "ключ" (обмен Диффи-Хеллмана) между двумя участниками сеанса связи при инициировании соединения. Затем этот общий "ключ" используется для обеспечения защиты (набора) мультимедийных ключей, которые используются для шифрования сеансов мультимедийной связи (RTP).

Профиль защиты на основе шифрования речевых сообщений является факультативным вариантом расширения для базового профиля защиты и для профиля защиты на основе подписи; его использование может быть оговорено при согласовании характеристик защиты терминала. В условиях, когда конфиденциальность речевых сообщений достигается другим способом, нет никакой необходимости применять шифрование мультимедийной информации и связанные с этим процедуры управления ключами (соглашение по ключам Диффи-Хеллмана, обновление ключей и синхронизация).

К выбранным алгоритмам шифрования относятся – AES, алгоритм, совместимый с RC2, DES и тройной DES.

ПРИМЕЧАНИЕ. – Поскольку для алгоритма DES можно также использовать вариант тройного DES, таким образом обеспечивается компактная реализация алгоритма.

Независимо от того, какой конкретный алгоритм шифрования мультимедийной информации выбран, необходимо четкое соответствие следующим опциям:

- Вектор инициализации (IV) генерируется, если он необходим, как определено в B.3.1.
- Заполнение (при необходимости) должно производиться, как это описано в B.3.2.

Полезная аудионагрузка должна шифроваться с использованием согласованного алгоритма шифрования ("X", "Y", "Z3" или "Z"), согласно процедурам, описанным в пункте 11 и Приложении B и в соответствии с методами заполнения зашифрованного текста, описанными в I.1. Полезная аудионагрузка может шифроваться с использованием согласованного алгоритма шифрования ("X1", "Y1", "Z1" или "Z2"), действующего в режиме поточного шифрования (EOFB).

D.7.1 Управление ключами

Конечные точки, соответствующие этому приложению должны применять процедуру быстрого соединения согласно 8.6.1. Если не применяется быстрый старт, то тогда надо использовать туннелирование H.245 для защиты сообщений управления вызовом H.245 с помощью этого приложения. Процедуры быстрого старта позволяют вводить один или два однонаправленных логических канала. Процедура быстрого старта предусматривает согласование характеристик защиты для распределения общего секретного "ключа" (общий DH "ключ"), который действует как главный ключ, и для безопасного распределения ключа шифрования.

В таблице D.4 представлены присвоенные OID для различных алгоритмов шифрования, которые соотносятся с OID, присвоенными группе Диффи-Хеллмана. Три группы DH идентифицируются посредством OID:

- "DHdummy": Экземпляр этой группы DH должен будет применяться тогда, когда речь идет об экспортируемой защите (512 битов) или когда используется любая из нестандартных групп DH.
- ПРИМЕЧАНИЕ 1. – Никакая конкретная группа DH не определяется; OID указывает на любую нестандартную группу DH.
- Экземпляр 512-битовой группы DH должен будет использоваться для формирования главного ключа для распределения сеансового ключа(ей) при RC2-совместимом ("X") алгоритме шифрования или при 56-битовом DES ("Y").
- "DH1024": Эта группа DH должна применяться тогда, когда речь идет о высоком уровне безопасности (1024 бита). OID указывает на стандартизированную фиксированную группу DH. Эта группа DH должна будет использоваться для формирования главного ключа для распределения сеансового ключа(ей) при алгоритме шифрования тройной-DES ("Z").
- "DH1536": Эта группа DH предлагается в качестве опции для конечных точек версии 3, имеющих очень высокие требования к защите, превышающие защиту 1024-битовой группы DH. OID указывает на фиксированную группу DH. Эта группа DH должна будет использоваться для формирования главного ключа для распределения сеансового ключа(ей) при алгоритмах шифрования тройной-DES ("Z", "Z1") или AES-128 ("Z2", "Z3").

Рекомендуется применение определенной 1024-битовой или, что необязательно, 1536-битовой группы DH, если только другие требования защиты не приведут к предпочтению других параметров Диффи-Хеллмана. Кроме того, рекомендуется рассмотреть вопрос применения OID, идентифицирующих группы DH, см. 8.8. Тем не менее, при реализации надо быть готовым к получению точных параметров групп DH без явной индикации OID. В этом случае, при реализации необходимо будет удостовериться, что правильная группа DH передается в соответствии с таблицей D.4.

Конечные точки могут использовать параметры нестандартных групп DH. Использование OID "DHdummy" должно будет указывать на эти нестандартные группы DH. Вызываемая сторона должна будет сама решать, принимать ли такие группы DH.

ПРИМЕЧАНИЕ 2. – Выбор группы DH не исключает потребность в согласовании фактического алгоритма шифрования мультимедийной информации. Оно должно сопровождаться процедурой согласования характеристик терминала H.245.

ПРИМЕЧАНИЕ 3. – При применении алгоритма шифрования во время установления соединения (SETUP-to-CONNECT) не стоит использовать OID для индикации экземпляра группы Диффи-Хеллмана.

Таблица D.4/H.235 – Группы Диффи-Хеллмана

OID алгоритма шифрования	**DH-OID**	**Описание групп Диффи-Хеллмана**
"X", "X1" (совместимый с RC2), "Y", "Y1" (DES)	"DHdummy"	Mod-P, любой подходящий 512-битовый исходный ключ
"Z", "Z1" (тройной-DES), "Z2", "Z3" (AES)	"DH1024"	Mod-P, 1024-битовый исходный ключ Исходный ключ = $2^{1024} - 2^{960} - 1 + 2^{64} \times \{ [2^{894}\ \text{pi}] + 129093 \}$ = (179769313486231590770839156793787453197860296048756011706444 42368419718021615851936894783379586492554150218056548598050 3 64644054819923910005079287700335581663922955313623907650873 5 75991482257486257500742530207744771258955095793778424442426 61733472762929938766870920560605027081084290769293201912819 4 467627007)$_{10}$ Generator(Примечание) = 2
"Z", "Z1" (тройной-DES), "Z2", "Z3" (AES)	"DH1536"	Mod-P, 1536-битовый исходный ключ Исходный ключ = $2^{1536} - 2^{1472} - 1 + 2^{64} * \{ [2^{1406}\ \text{pi}] + 741804 \}$ = (241031242692103258855207602219756607485695054850245994265411 69419581088316826122288900938582613416146732271414779040121 9 65036489570505826319427307068050092230627347453410734066962 4 60145893616597740410271692494532003787294341703258437786591 9 81437631937768598695240889401955773461198435453015470437472 0 77499697637500843089263392955599688824578724129938101291302 9 45929999479263652640592846472097303849472116814344647144384 8 85209401274598442888593365268963209196339 19)$_{10}$ Generator (Примечание) = 2
ПРИМЕЧАНИЕ. – Generator используется для создания DH-маркера.		

D.7.2 Обновление и синхронизация ключей

При 64-битовом блочном шифре частота обновления ключей *должна* быть такой, чтобы с использованием одного и того же ключа зашифровалось не более, чем 2^{32} блоков. В конкретных реализациях ключи *должны* обновляться прежде, чем один и тот же ключ использовался для шифрования 2^{30} блоков (см. 11.1). При 128-битовом блочном шифре частота обновления ключей *должна* быть такой, чтобы с использованием одного и того же ключа шифровалось не более, чем 2^{64} блоков. В конкретных реализациях ключи *должны* обновляться до использования одного и того же ключа для шифрования 2^{62} блоков (см. 11.1). Оба участвующих объекта имеют право менять мультимедийный сеансовый ключ так часто, как этого требует их стратегия защиты. Например, ведущий объект может распределять новый сеансовый ключ, используя **encryptionUpdate** или **encryptionUpdateCommand** сообщения **miscellaneousCommand**. С другой стороны, ведомый объект может запросить новый сеансовый ключ от ведущего, используя **encryptionUpdateRequest** сообщения **miscellaneousCommand**, см. также B.2.6.

Сообщение **MiscellaneousCommand** содержит **encryptionUpdate** и **encryptionUpdateCommand**, **encryptionSynch** которых устанавливается со следующими параметрами:

- **synchFlag**: новый динамический номер полезной нагрузки RTP, указывающий на смену ключа.
- **h235key**: передает новый зашифрованный сеансовый ключ. Это закодированная в ASN.1 структура **H235Key** H.235, передаваемая в виде цепочки октетов.

Поле **sharedSecret** в рамках структуры **H235Key** использует следующие поля:

- **algorithmOID**: устанавливается в "X", "X1"для 56-битового совместимого с RC2 алгоритма, устанавливается в "Y", "Y1" для 56-битового DES или устанавливается в "Z", "Z1" для 168-битового тройного DES или же устанавливается в "Z3" для 128-битового AES.

  ПРИМЕЧАНИЕ 1. – Алгоритм шифрования сеансового ключа является тем же самым, что и согласованный алгоритм шифрования мультимедийной информации.
- **paramS**: устанавливается в исходное значение. При 64-битовом блочном шифре **iv8** содержит случайную комбинацию 64-битовых блоков, которую формирует инициатор. При 128-битовом блочном шифре **iv16** содержит случайную 128-битовую комбинацию блоков, которую формирует инициатор. Это поле не используется при режиме CBC и должно принимать значение NULL, при условии, что CBC-IV для шифрования сеансового ключа должен быть установлен в 0; он должен использоваться только для переноса IV в режиме EOFB.
- **encryptedData**: имеет своим значением результат шифрования **KeySyncMaterial**.

В составе **KeySyncMaterial**:

- **generalID**: идентификатор источника, распределяющего ключ.

  ПРИМЕЧАНИЕ 2. – В данной Рекомендации предполагается, что каждая конечная точка была зарегистрирована контроллером доступа и получила идентификатор конечной точки, который может быть передан в рамках **generalID**. Данная Рекомендация не поддерживает сценарии без контроллеров доступа, такой вариант подлежит дальнейшему изучению.
- **keyMaterial**: устанавливается равным новому сеансовому ключу. Для алгоритмов DES и алгоритма, совместимого с RC2, им является 56-битовый ключ, для тройного DES им является 168-битовый ключ, а для AES – 128-битовый ключ. Ведущий должен будет сформировать новый сеансовый ключ, который соответствует, по крайней мере, следующим критериям защиты: он не должен быть неустойчивым или отчасти неустойчивым DES-ключом и он должен использовать достаточно защищенный случайный источник.

Сообщение **MiscellaneousCommand** содержит **encryptionUpdateRequest**, который содержит **keyProtectionMethod**, где флаг **sharedSecret** установлен в TRUE.

ПРИМЕЧАНИЕ 3. – Поскольку обновление и синхронизация ключей зависят от сообщений H.245, которые обратно несовместимы во время быстрого соединения, то для защищенных объектов H.323 требуется туннелирование сообщений H.245.

D.7.3 Тройной DES во внешнем режиме CBC

Тройной 168-битовый DES во внешнем режиме CBC, как показано на рисунке D.4, *должен* использоваться в пределах этого профиля защиты. На этом рисунке каждый k_i относится к 56-битовому ключу. Разные 56-битовые ключи *должны* будут использоваться в каждом блоке шифрования (E) и дешифрования (D). Ни один из 64 неустойчивых ключей для DES, судя по всему, не вызывает никакой неустойчивости в тройном DES. Однако реализации, соответствующие этому профилю, при задействовании неустойчивого ключа DES должны отвергать этот ключ (см. RFC 2405).

Дополнительные сведения о тройном DES можно получить из [Schneier] и RFC 2405.

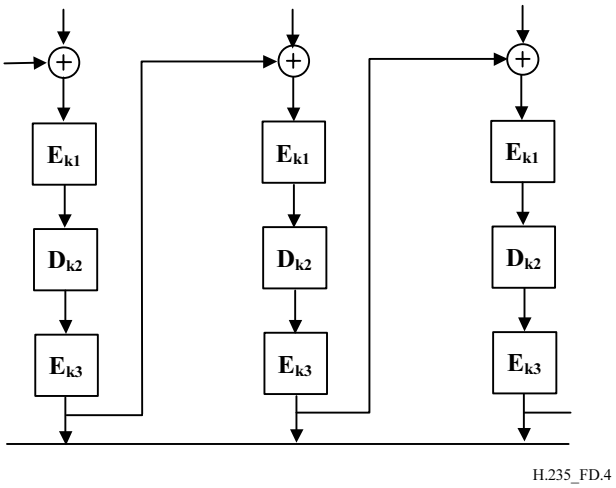


Рисунок D.4/H.235 – Шифрование в тройном DES во внешнем режиме CBC

D.7.4 Алгоритм DES, действующий в режиме EOFB

Речевые сообщения могут кодироваться с использованием алгоритма DES действующего в режиме поточного шифрования сцеплений блоков в режиме EOFB. Режим EOFB позволяет применять принцип параллелизма при реализациях. При работе в режиме EOFB рекомендуется, по причинам безопасности и эксплуатационного качества, возвратить к источнику весь криптоблок (то есть, все 64 бита в DES при, к примеру, $n = j = 64$). Однако, вследствие того, что режим EOFB не обеспечивает сцепление блоков и битов, он может быть чувствительным к конкретным попыткам нарушения защиты, в зависимости от статистических свойств входных нешифрованных данных. Так что, обновление ключей (см. D.7.2) должно происходить регулярно и, по крайней мере, до свертывания исходных значений. По вопросу вычисления исходного значения см. B.3.1.2.

D.7.5 Тройной DES во внешнем режиме EOFB

168-битовый тройной DES во внешнем режиме EOFB, как показано на рисунке D.5, может использоваться в рамках данного профиля защиты. На этом рисунке каждая величина k_i соответствует 56-битовому ключу. В рамках каждого блока шифрования (E) и дешифрования (D) *должен* будет использоваться другой 56-битовый ключ. Отсутствуют данные о том, что какой-либо из 64-битовых неустойчивых ключей при DES послужил причиной неустойчивости в рамках тройного DES. Тем не менее, реализации, соответствующие этому профилю, должны отвергать этот ключ, если задействован неустойчивый ключ DES [RFC 2405].

Дополнительную информацию о тройном DES можно получить в [Schneier] [RFC 2405].

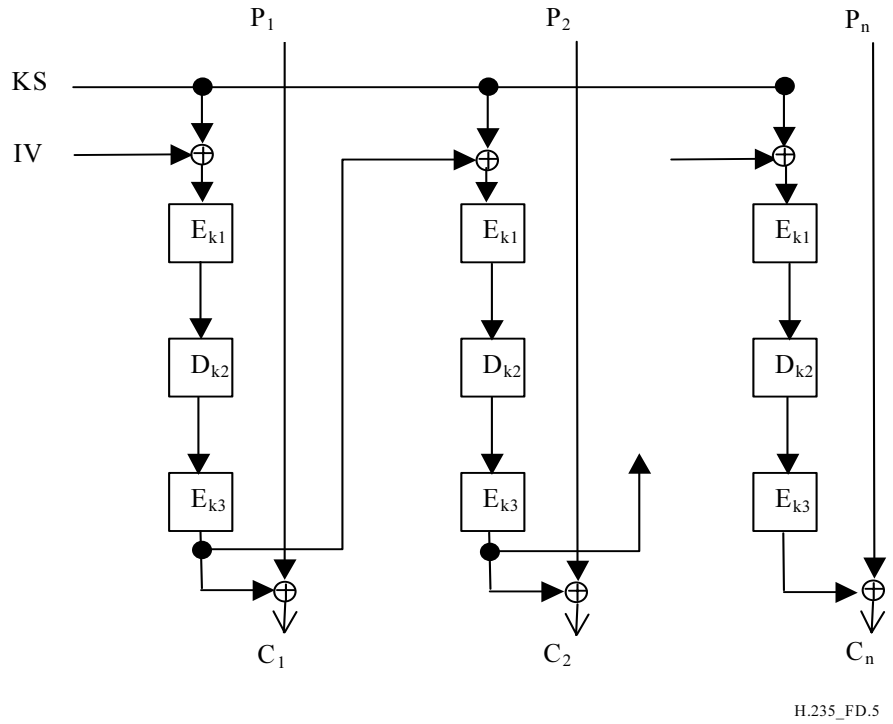


Рисунок D.5/H.235 – Тройной DES во внешнем режиме EOFB

D.8 Санкционированный перехват

Этот вопрос подлежит дальнейшему изучению (см. [LI]).

D.9 Перечень защищенных сигнальных сообщений

В настоящем пункте кратко описано, как и посредством каких средств в Приложении D обеспечивается защита различных сигнальных сообщений H.323.

D.9.1 Сообщение RAS H.225.0

Сообщение RAS H.225.0	Сигнальные поля H.235	Аутентификация и контроль целостности
Любое	cryptoTokens	Процедура I

D.9.2 Передача сигналов вызова H.225.0

Сообщение о посылке вызова H.225.0	Сигнальные поля H.235	Аутентификация и контроль целостности
Alerting-UUIE, CallProceeding-UUIE, Connect-UUIE, Setup-UUIE, Facility-UUIE, Progress-UUIE, Information-UUIE, ReleaseComplete-UUIE, Status-UUIE, StatusInquiry-UUIE, SetupAcknowledge-UUIE, Notify-UUIE	cryptoTokens	Процедура I

D.9.3 Управление вызовом H.245

Сообщения H.245, передаваемые и принимаемые защищенными объектами H.323, должны будут либо иметь возможность обратного совмещения составной части защищенного быстрого соединения, либо они должны будут туннелироваться с использованием защищенного сообщения **Facility-UUIE** H.225.0.

D.10 Использование sendersID и generalID

ClearToken содержит поля **sendersID** и **generalID**. При наличии идентификационной информации **sendersID** должны быть установлены в значение идентификатора контроллера доступа (GKID) для

инициированного этим контроллером доступа сообщения и в значение идентификатора конечной точки (EPID) для сообщений, инициированных конечной точкой. При наличии идентификационной информации **generalID** должен быть установлен равным GKID для сообщений, инициированных конечной точкой, и должен будет иметь значение EPID для сообщений, инициированных контроллером доступа. Когда идентификационная информация недоступна или в случае, когда результат широковещательной или многоадресной передачи неоднозначен, поле пропускается или должно будет содержать нулевую строку. В таблице D.5 дано краткое описание этой ситуации.

Таблица D.5/H.235 – Идентификаторы объектов, используемые в Приложении D

Сообщение	sendersID	generalID
Одноадресное **GRQ**	**EPID**, если доступен, иначе **NULL**	**GKID**
Многоадресное **GRQ**	**EPID**, если доступен, иначе **NULL**	
GCF, **GRJ**	**GKID**	**EPID**, если доступен, иначе **NULL**
Исходное **RRQ**	**EPID**, если доступен, иначе **NULL**	**GKID**
RCF	**GKID**	**EPID**
RRJ	**GKID**	
URQ, **UCF**, **URJ**, **BRQ**, **BCF**, **BRJ**, **DRQ**, **DCF**, **DRJ**, **NSM**, **RIP**, **SCI**, **SCR**, **XRS** (от EP к GK)	**EPID**	**GKID**
URQ, **UCF**, **URJ**, **BRQ**, **BCF**, **BRJ**, **DRQ**, **DCF**, **DRJ**, **NSM**, **RIP**, **SCI**, **SCR**, **XRS** (от GK к EP)	**GKID**	**EPID**
ARQ, **IRQ**, **RAI**	**EPID**	**GKID**
ACF, **ARJ**, **BCF**, **LCF**, **LRJ**, **IRR**, **IRQ**, **RAC**, **LCF**, **LRJ**, **IACK**, **INAK**	**GKID**	**EPID**
Одноадресное **LRQ** (от EP к GK)	**EPID**	**GKID**
Одноадресное **LRQ** (от GK к GK)	**GKID**	**GKID**
Многоадресное **LRQ**	**EPID**	
ПРИМЕЧАНИЕ. – GKID обозначает идентификатор контроллера доступа. EPID обозначает идентификатор конечной точки. Пробел указывает на отсутствие или на нулевую идентификационную строку.		

D.11 Перечень идентификаторов объектов

В таблице D.6 представлены все упомянутые OID (см. также [OIW] и [WEBOID]). Даны идентификаторы объектов для H.235v1 [H.235 v1] и для H.235v2 [H.235 v2].

Таблица D.6/H.235 – Идентификаторы объектов, используемые в Приложении D

Название идентификатора объекта	Значение идентификаторов объектов	Описание
"A"	{itu-t (0) recommendation (0) h (8) 235 version (0) 2 1} {itu-t (0) recommendation (0) h (8) 235 version (0) 1 1}	Используется в Процедуре I для CryptoToken-tokenOID, указывая, что хеш-величина включает все поля сообщения RAS H.225.0 и сообщения о посылке вызова (аутентификация и контроль целостности)
"E"	{itu-t (0) recommendation (0) h (8) 235 version (0) 3 9} {itu-t (0) recommendation (0) h (8) 235 version (0) 2 9}	Сквозной ClearToken, переносящий sendersID для верификации на стороне получателя
"T"	{itu-t (0) recommendation (0) h (8) 235 version (0) 2 5} {itu-t (0) recommendation (0) h (8) 235 version (0) 1 5}	Используется в Процедуре I и IA в качестве основного ClearToken для аутентификации и защиты от повторных попыток нарушения защиты, и факультативно, также для управления ключами Диффи-Хеллмана, как описано в D.7.1
"U"	{itu-t (0) recommendation (0) h (8) 235 version (0) 2 6} {itu-t (0) recommendation (0) h (8) 235 version (0) 1 6}	Используется в Процедуре I для OID алгоритма, указывая на использование HMAC-SHAl-96
"DHdummy"	{itu-t (0) recommendation (0) h (8) 235 version (0) 2 40} {itu-t (0) recommendation (0) h (8) 235 version (0) 3 40}	Явно предоставляемая нестандартная DH-группа
"DH1024"	{itu-t (0) recommendation (0) h (8) 235 version (0) 2 43} {itu-t (0) recommendation (0) h (8) 235 version (0) 3 43}	1024-битовая DH-группа
"DH1536"	{itu-t (0) recommendation (0) h (8) 235 version (0) 3 44}	1536-битовая DH-группа
"X"	{iso(1) member-body(2) us(840) rsadsi(113549) encryptionalgorithm(3) 2}	Шифрование речевых сообщений, с использованием совместимого с RC2 алгоритма (56 битов) или совместимого с RC2 алгоритма в режиме CBC и 512-битовой DH-группы
"X1"	{itu-t (0) recommendation (0) h (8) 235 version (0) 3 27}	Шифрование речевых сообщений, с использованием совместимого с RC2 алгоритма (56 битов) или совместимого с RC2 алгоритма в режиме EOFB и 512-битовой DH-группы.
"Y"	{iso(1), identified-organization(3), oiw(14), secsig(3), algorithm(2), descbc(7)}	Шифрование речевых сообщений с использованием DES (56 битов) в режиме CBC и 512-битовой DH-группы
"Y1"	{itu-t (0) recommendation (0) h (8) 235 version (0) 3 28}	Шифрование речевых сообщений с использованием DES (56 битов) в режиме EOFB и 512-битовой DH-группы с 64-битовой обратной связью

Таблица D.6/H.235 – Идентификаторы объектов, используемые в Приложении D

Название идентификатора объекта	Значение идентификаторов объектов	Описание
"Z1"	{itu-t (0) recommendation (0) h (8) 235 version (0) 3 29}	Шифрование речевых сообщений с использованием тройного-DES (168 битов) во внешнем режиме EOFB и 1024-битовой DH-группы с 64-битовой обратной связью
"Z2"	{itu-t (0) recommendation (0) h (8) 235 version (0) 3 30}	Шифрование речевых сообщений, с использованием AES (128 битов) в режиме EOFB и 1024-битовой DH-группы
"Z3"	{joint-iso-itu-t(2) country(16) us(840) organization(1) gov(101) 3 nistAlgorithm(4) aes(1) cbc(2)}	Шифрование речевых сообщений с использованием AES (128 битов) в режиме CBC и 1024-битовой DH-группы
"Z"	{iso(1) identified-organization(3) oiw(14) secsig(3) algorithm(2) desEDE(17)}	Шифрование речевых сообщений с использованием тройного DES (168 битов) во внешнем режиме CBC и 1024-битовой DH-группы.

D.12 Библиография

[FIPS PUB 180-1] NIST, FIPS PUB 180-1: Стандарт аутентификации и проверки целостности информации, апрель 1995 г.; http://csrc.nist.gov/fips/fipl80-l.ps

[LI] Проект DRT/TIPHON-08003 VO.0.9, "Санкционированный перехват – Внутренний интерфейс LI", август 2000 г.

[OIW] Устойчивая реализация – Соглашения по протоколам Взаимодействия Открытых Систем: Часть 12 – Защита ОС; Выпуск от декабря 1994. Рабочая группа разработчиков среды открытых систем (OIW); http://nemo.ncsl.nist.gov/oiw/agreements/stable/OSI/12s_9412.txt

[RFC 2405] С. Мэдсон, Н. Дорасвами "ESP DES-CBC Алгоритм шифрования с явным IV", RFC 2405, Целевая группа проектирования инженерной поддержки сети Интернет, 1998 г.

[WEBOID] http://www.alvestrand.no/objectid/top.html

Приложение E

Профиль защиты в виде подписи

E.1 Общее описание

В данном Приложении описывается предлагаемый в качестве опции профиль защиты, использующий цифровые подписи. Защищенные объекты H.323 (терминалы, контроллеры доступа, шлюзы, MCU и т. д.) могут применять этот профиль защиты в виде подписи для усовершенствования защиты или по мере необходимости.

Профиль защиты в виде подписи позволяет применять модель маршрутизации с помощью GK, он основывается на методах туннелирования H.245; вопрос поддержки других моделей, отличных от модели маршрутизации с помощью GK, подлежит дальнейшему изучению.

Профиль защиты в виде подписи применим для масштабируемой "глобальной" IP-телефонии; этот профиль защиты снимает ограничения, присущие простым базовым профилям защиты из Приложения D. Например, профиль защиты в виде подписи не зависит от организации общих "ключей" для сегментов

сети в различных доменах. Он обеспечивает туннелирование сообщений H.245 для контроля целостности сообщений H.245 и, кроме того, создает условия для защиты от неподтверждения сообщений. Профиль защиты в виде подписи поддерживает посегментную защиту, а также фактическую сквозную аутентификацию с одновременным использованием прокси H.235 или промежуточных контроллеров доступа.

К свойствам, обеспечиваемым этими профилями для сообщений RAS, H.225.0 и H.245, относятся следующие:

- Аутентификация пользователя нужным объектом, независимо от количества сетевых сегментов[7] прикладного уровня, которые проходит сообщение.
- Сохранение целостности всех или критических частей (полей) сообщений, поступающих к объекту, независимо от количества сетевых сегментов прикладного уровня, которые проходит сообщение. Само обеспечение целостности сообщения на основе использования созданного действительно случайного числа также необязательно.
- Посегментная аутентификация сообщений прикладного уровня, контроль целостности сообщений и защита от неподтверждения обеспечивают сетевые средства защиты для всего сообщения.
- Также может обеспечиваться защита от неподтверждения в сообщениях, которыми обмениваются два объекта, независимо от числа сетевых сегментов прикладного уровня, которые проходит сообщение. А именно, защита от неподтверждения обеспечивается для критических частей (полей) сообщения. Например, это может иметь место, когда EP посылает сообщение SETUP своему GK, а два объекта (EP и GK) отделены одним или несколькими прокси.

Для предотвращения некоторых попыток нарушения защиты используются вышеуказанные варианты сетевых средств защиты. К ним относятся:

- Попытки нарушения защиты типа "отказ в обслуживании": Быстрая проверка цифровых подписей может предотвратить такие попытки.
- Попытки нарушения защиты со стороны посторонних лиц: Последовательная (посегментная) аутентификация сообщений на прикладном уровне и контроль их целостности препятствуют подобным попыткам, когда посторонний объект, например, "недружелюбный" маршрутизатор, находится между сегментами прикладного уровня. Если такая попытка производится объектом прикладного уровня, то такие попытки предотвращаются посредством введения сквозной аутентификации пользователей и контролем целостности для выбранных частей сообщения.
- Повторные попытки нарушения защиты: Использование временных меток и порядковых номеров предотвращают такие попытки.
- Спуфинг (имитация соединения): Аутентификация пользователя предотвращает такие попытки.
- "Захват" соединения: Применение аутентификации/контроля целостности для каждого сигнального сообщения предотвращает такие попытки.

E.2 Термины, принятые в спецификациях

При необходимости, для достижения конфиденциальности речевых сообщений профиль защиты в виде подписи может использовать **профиль защиты на основе шифрования речевых сообщений**, описанный в Приложении D.

Процедуры II и III определяют способ реализации сетевых средств защиты для таких различных сценариев, как посегментная защита и сквозная защита с использованием различных механизмов защиты типа асимметричных криптографических методов (цифровой подписи).

При том, что сетевое средство контроля целостности сообщений всегда осуществляет и аутентификацию сообщений, обратное утверждение не всегда справедливо. При режиме только – аутентификации контроль целостности распространяется только на определенное подмножество полей сообщения. Это применимо к сетевым средствам, обеспечивающим контроль целостности, реализуемым с помощью асимметричных методов (например, цифровые подписи). На практике объединенные сетевые средства аутентификации и контроля целостности используют одни и те же данные ключа, и это не ослабляет защиту.

---

[7] "Сегмент" понимается здесь как доверительный сетевой элемент H.235 (например, GK, GW, MCU, прокси, брандмауэр). Таким образом, посегментная защита прикладного уровня с применением симметричных методов не обеспечивает фактическую сквозную защиту между терминалами.

Кроме того, вся информация о посегментной защите сети помещается в элемент **CryptoSignedToken**. Эта информация заново вычисляется на каждом сегменте согласно Процедуре II.

С другой стороны, информация о сквозной защите, обеспечиваемая только при использовании прокси H.323 и Процедуры III, рассчитывается на основе той информации, что находится в **CryptoSignedToken**, но хранится эта информация в отдельном **CryptoToken** сообщения. Эта информация не изменяется при пересылке. Отдельный идентификатор объекта позволяет различать посегментный **CryptoToken** и сквозной **CryptoToken**.

Сертификационный центр: Если Сертификационный центр (CA) указывается в контексте электронной подписи, то он удостоверяет открытые ключи, прошедшие верификацию, выдавая "Сертификаты".

Архивы сертификатов: Архивы сертификатов (например, каталог X.500) содержат сертификаты пользователей и Перечни отмененных сертификатов (CRL). Им доверено представление этой информации, но они не отвечают за содержание или точность информации, которую они получают от CA или от RA.

Цифровая подпись: Является криптографической трансформацией (используя асимметричный криптографический метод) численного представления сообщения, содержащего данные, по которым любое лицо, имеющее подписанное сообщение и соответствующий открытый ключ, может определить, что:

i) эта трансформация была произведена с применением личного ключа, соответствующего определенному открытому ключу; и

ii) подписанное сообщение не изменялось, начиная с момента криптографической трансформации.

Провайдеры оперативного статуса сертификата: Протокол оперативного статуса сертификата (OCSP) дает возможность приложениям определять статус отмены идентифицированного сертификата. OCSP может использоваться для того, чтобы удовлетворять некоторые из эксплуатационных требований к обеспечению информации об отмене более оперативно, чем это возможно с помощью CRL. Провайдеров оперативного статуса сертификата можно считать альтернативой оперативному использованию CRL.

Прокси (модуль-посредник): Прокси является промежуточным объектом H.323, подобным контроллеру доступа. Прокси может быть отдельным сетевым узлом или он может быть совмещен по выполняемым функциям с таким объектом H.323, как контроллер доступа. Прокси может выполнять задачи защиты, проверку подписи и сертификата, и управление доступом.

Регистрационные центры: Регистрационные центры действуют как посредники между потребителями и сертификационными центрами (CA). Они принимают запросы от пользователей и передают их к CA соответствующим образом.

Центры выдачи временных меток: Центры выдачи временных меток необходимы для защиты от неподтверждения в случае потери или несанкционированного раскрытия ключа. На практике, они обеспечивают подпись, удостоверяющую другую подпись, для любого объекта, включая надежное время, посредством хеш-величины и хеш-идентификатора.

Доверенный провайдер услуг: Объект, который может использоваться другими объектами как доверенный посредник в процессе связи или в процессе проверки или же, как доверенный провайдер информационных услуг.

Профиль защиты в виде подписи предлагается в качестве опции. Такой профиль защиты применим в условиях с потенциально большим количеством терминалов, причем присвоение пароля/симметричного ключа невозможно, к примеру, в крупномасштабных или глобальных сценариях. Профиль защиты в виде подписи обеспечивает дополнительные сетевые средства защиты для защиты от неподтверждения с использованием цифровых подписей и сертификатов. Цифровые подписи могут использовать функцию хеширования на основе SHA1 или MD5 и обеспечивают аутентификацию и/или контроль целостности (см. Процедуры II и III).

Объекты H.323, использующие аутентификацию и контроль целостности или только-аутентификацию на посегментной основе, должны будут использовать Процедуру II. Объекты H.323, выполняющие только-аутентификацию, не должны проводить контроль целостности. Объекты H.323, выполняющие только-аутентификацию должны будут использовать для фактической сквозной аутентификации Процедуру III.

В данном приложении применима защита целостности сообщений, которая охватывает полное сообщение. Для RAS H.225.0 защита целостности охватывает полное сообщение RAS; для передачи сигналов вызова она охватывает полное сообщение о посылке вызова H.225.0, включая заголовки Q.931.

Профиль защиты в виде подписи позволяет надежно туннелировать PDU управления вызовом H.245 в сообщениях **facility** H.225.0. Механизмы обновления и синхронизации ключей H.245 требуют туннелирования, что, например, удобно при вызовах с очень большой продолжительностью[8].

Вертикально заштрихованная область (желтая – в электронной версии) в таблице E.1 соответствует области действия профиля защиты в виде подписи. Если контроль целостности пропущен, что обозначено горизонтально заштрихованной областью (голубая – в электронной версии), то результатом является профиль защиты с только-аутентификацией. Дополнительной возможностью при использовании профиля защиты в виде подписи является выбор между цифровыми подписями RSA-SHA-1 или RSA-MD5. Профиль защиты на основе шифрования речевых сообщений, описанный в Приложении D (см. D.7), может дополнительно использоваться вместе с профилем защиты в виде подписи.

Таблица E.1/H.235 – Профиль защиты в виде подписи

Сетевые средства защиты	**Функции вызова**			
	RAS	**H.225.0**	**H.245**[a)]	**RTP**
Аутентификация	SHA1/ MD5	SHA1/ MD5	SHA1/ MD5	
	цифровая подпись	цифровая подпись	цифровая подпись	
Защита от неподтверж-дения	SHA1/ MD5	SHA1/ MD5	SHA1/ MD5	
	цифровая подпись	цифровая подпись	цифровая подпись	
Контроль целостности	SHA1/ MD5	SHA1/ MD5	SHA1/ MD5	
	цифровая подпись	цифровая подпись	цифровая подпись	
Конфиден-циальность				
Управление доступом				
Управление ключами	Распределение сертификатов	Распределение сертификатов		

a) Туннелированное H.245 или встроенное H.245 в рамках быстрого соединения H.225.0.

ПРИМЕЧАНИЕ 1. – Профиль защиты в виде подписи должен также поддерживаться другими объектами H.235 (например, контроллерами доступа, шлюзами и прокси H.235).

ПРИМЕЧАНИЕ 2. – Имеющиеся в сертификате биты использования ключей могут также определять, какое сетевое средство защиты обеспечивается терминалом (например, заявленная защита от неподтверждения).

Для аутентификации потребитель должен использовать схему подписи на основе открытых или личных ключей. Такая схема обычно обеспечивает лучший контроль целостности и защиту от неподтверждения при вызове.

Настоящая Рекомендация не описывает процедуру для:

- регистрации, сертификации и распределения сертификатов от доверительного центра, а также – присвоения открытых или личных ключей, служб каталогов, конкретных параметров CA, отмены сертификатов, обновления или восстановления пар ключей и других процедур работы с

---

[8] Обновление ключей для защищенного кодирования речи согласно G.711 должно происходить, самое позднее, после передачи 2^{30} 64-битовых блоков, что составляет более 12 дней непрерывного разговора.

сертификатами или управления ими, таких как, доставка сертификатов и открытых/личных ключей и реализация их в терминалах.

Такие процедуры могут потребовать использования технических средств, не рассматриваемых в настоящем Приложении.

Участвующие в процессе связи объекты способны неявно определить использование или базовых профилей защиты, описанных в Приложении D, или данного профиля защиты в виде подписи, путем определения сигнализирующих защиту идентификаторов объектов в сообщениях (**tokenOID** и **algorithmOID**; см. также E.18).

E.3 Требования к объектам H.323

Предполагается, что объекты H.323, которые реализуют этот профиль в виде подписи, поддерживают следующие свойства H.323:

- быстрое соединение;
- модель маршрутизации с помощью контроллера доступа.

E.4 Сетевые средства защиты

В настоящем Приложении, применительно к обеспечению сетевых средств защиты, используются следующие термины.

- **Только-аутентификация**: Это сетевое средство защиты, имеющее профиль защиты в виде подписи, обеспечивает аутентификацию пользователя, в ходе которой пользователь аутентифицирует по правильной цифровой подписи какую-то часть данных, используя личный ключ. Отметим, что это сетевое средство защиты не обеспечивает противодействие случайным операциям вырезания и вставки информации, манипулирования сообщениями или попыткам искажения информации. Только-аутентификация может быть полезна для защитных прокси, которые проверяют подлинность сообщения (аутентификация источника данных) при переадресации[9] сообщения к другому назначению (например, контроллеру доступа). Тем не менее, только-аутентификация может также применяться посегментно. Процедура III определяет это сетевое средство защиты для сквозного сценария, в то время, как Процедура II определяет это сетевое средство защиты для посегментного сценария.
- **Аутентификация и контроль целостности**: Это объединенное сетевое средство защиты, которое поддерживает контроль целостности сообщения вместе с аутентификацией пользователя. Пользователь аутентифицируется по правильной цифровой подписи какой-либо части данных, используя личный ключ. В дополнение к этому, сообщение защищено от искажения информации. Оба сетевых средства защиты имеют один и тот же механизм защиты. Объединенные аутентификация и контроль целостности возможны только на посегментной основе. Процедура II определяет это сетевое средство защиты.

ПРИМЕЧАНИЕ. – Когда применяются цифровые подписи, может поддерживаться сетевое средство защиты в виде защиты от неподтверждения; это также зависит от установок битов использования ключа подписи в сертификате (см. также RFC 3280).

Асимметричные методы, использующие цифровые подписи, могут применяться в сценариях посегментной и/или также сквозной передачи сообщений.

Ниже описаны процедуры, используемые в этом профиле:

Процедура II основывается на цифровых подписях с использованием пары личных/открытых ключей для обеспечения аутентификации, контроля целостности и защиты от неподтверждения сообщений RAS, Q.931 и H.245. Терминалы могут использовать этот метод, если необходима защита от неподтверждения и сложный контроль целостности.

В зависимости от стратегии защиты, аутентификация может быть односторонней или же взаимной, при этом обеспечивается более высокая степень защиты и применение аутентификации/контроля целостности также и в обратном направлении. Стратегия защиты терминала может позволить "только-аутентификацию" без вычисления криптографической целостности (см. E.7).

Контроллеры доступа, обнаружившие безуспешное подтверждение аутентификации и/или безуспешное подтверждение целостности в сообщении RAS или сообщении о посылке вызова, полученном от терминала или однорангового контроллера доступа, реагируют соответствующим сообщением отказа,

---

[9] Переадресовка обычно изменяет некоторые части сообщения; поэтому сквозная целостность не может быть обеспечена.

указывающим на безуспешность защиты путем установки причины отказа в **securityDenial** или в другой код ошибки защиты согласно B.2.2. В зависимости от способности распознавания попыток нарушения защиты и наиболее подходящего способа реагирования на эти попытки, контроллер доступа, получающий защищенное **xRQ** с неопределенными идентификаторами объектов (**tokenOID**, **algorithmOID**), должен будет ответить незащищенным **xRJ** или же он может сбросить это сообщение. Возникающее событие защиты должно регистрироваться. С другой стороны, конечная точка должна будет сбросить полученное незащищенное сообщение, сделать перерыв и снова повторить попытку путем выбора разных OID. Подобным же образом, контроллер доступа, получивший защищенное сообщение SETUP H.225.0 с неопределенными идентификаторами объектов (**tokenOID**, **algorithmOID**), должен будет ответить незащищенным RELEASE COMPLETE с причиной, установленной в **securityDenied**, или же он может сбросить это сообщение. Точно также, возникающее событие защиты должно регистрироваться.

Присутствует неявная сигнализация H.235 для индикации использования Процедуры II и применяющегося механизма защиты, основывающаяся на значении идентификаторов объектов (см. также пункт E.18) и заполнении полей сообщения. Идентификаторы объектов в этом тексте обозначаются буквами (например, "A").

Этот профиль не использует поля ICV H.235; вместо них лучше поместить контрольные признаки криптографической целостности в поле **signature** элемента **token** в **cryptoSignedToken**.

E.5 Подробное описание цифровых подписей с парами открытых/личных ключей (Процедура II)

Если для обеспечения посегментной защиты используется Процедура II, то следует придерживаться следующих процедур:

- Для создания цифровой подписи должны использоваться алгоритмы SHA1 или MD5 вместе с алгоритмом RSA. Строгое соблюдение PKCS #1 и PKCS #7 содействует при этом функциональной совместимости.

  Поле **CryptoH323Token** в каждом сообщении RAS/H.225.0 должно будет содержать следующие поля:

  – **nestedCryptoToken**, содержащее **CryptoToken**, которое само содержит **cryptoSignedToken**, содержащее следующие поля:

  - **tokenOID**, установленное в:
    – "A", указывающее, что вычисление при аутентификации/контроле целостности учитывает все поля в сообщении RAS/H.225.0 или сообщении о посылке вызова (см. пункт E.9);
    – "B", указывающее, что вычисление при аутентификации/контроле целостности учитывает только подмножество полей (см. пункт E.8) сообщения RAS/H.225.0 для только-аутентификации.
  - **token**, содержащее поля:
    – **toBeSigned**, содержащее **EncodedGeneralToken**, которое фактически является **ClearToken** с набором следующих полей:
      - **tokenOID**, установленное в "S", которое указывает, что **ClearToken** используется для аутентификации/контроля целостности/защиты от неподтверждения;
      - **timeStamp**, содержащее временную метку;
      - **random**, содержащее равномерно возрастающий порядковый номер;
      - **generalID**, содержащее идентификатор получателя (только в случае одноадресных сообщений);
      - **sendersID**, содержащее идентификатор отправителя;
      - **dhkey**, используемое для передачи параметров Диффи-Хеллмана во время передачи сообщений **Setup**-to-**Connect**, как это определено в настоящей Рекомендации:
        – **halfkey**, содержащее случайный открытый ключ одного из участников;
        – **modsize**, содержащее исходный ключ DH (см. таблицу D.4);
        – **generator**, содержащее DH-группу (см. таблицу D.4).

        ПРИМЕЧАНИЕ 1. – Когда профиль защиты в виде подписи используется без профиля защиты на основе шифрования речевых сообщений, не нужно передавать

никакие параметры Диффи-Хеллмана, а **dhkey** должен отсутствовать; **halfkey**, **modsize** и **generator** могут быть установлены в {'0'B,'0'B,'0'B}.

- **certificate** содержит цифровой сертификат отправителя, причем тип указывает тип сертификата ("V" для сертификатов MD5-RSA или "W" для сертификатов SHA1-RSA), а **certificate** переносит фактический сертификат (см. E.12).
- **algorithmOID** устанавливается в:
  - "V", указывающее на использование подписи MD5-RSA;
  - "W", указывающее на использование подписи SHA1-RSA.
- **paramS** устанавливается в NULL.
- **signature** содержит подпись, вычисленную с использованием SHA1 или MD5 RSA по всем полям (если **tokenOID** – "A", см. E.9) или по определенным критическим полям (если **tokenOID** – "B", см. E.8) сообщения RAS H.225.0 или сообщения о посылке вызова.

Когда значение **tokenOID** "A" используется для защиты туннелированных H323-UU-PDU, включая все содержимое сообщений H.245, тогда вычисление подписи должно производиться по полному сообщению H.225.0 о посылке вызова со всеми полями, согласно процедуре, описанной в E.9. В том случае, если используется значение **tokenOID** "B", то при применении Процедуры III осуществляется только-аутентификация **CryptoToken** (см. E.8).

- Объект (который может находиться на расстоянии одного или нескольких сегментов прикладного уровня), для которого предназначается подпись, проверяет ее.

ПРИМЕЧАНИЕ 2. – Получатель способен обнаружить использование Процедуры II, проверяя **algorithmOID** в маркере **cryptoSignedToken** (обнаруживая присутствие "V" или "W").

E.6 Процедуры проведения многосторонней конференции

MCU должны будут поддерживать защищенное распределение сертификатов по запросу от терминалов посредством туннелирования команд H.245 **ConferenceRequest** и **ConferenceResponse**, как описано в 9.1. Это позволяет терминалам запрашивать сертификаты от других терминалов в условиях многосторонней конференции и, таким образом, удостовериться в идентичности других участников конференции.

ConferenceRequest передает **requestTerminalCertificate**, следующие поля которого установлены в:

- **terminalLabel**: используется как средство адресации удаленного терминала посредством MCU;
- **certSelectionCriteria**: отправитель может запрашивать сертификаты только конкретных типов;
- **sRandom**: случайный запрос, генерированный запрашивающим отправителем.

ConferenceResponse передает **terminalCertificateResponse**, следующие поля которого установлены в:

- **terminalLabel**: позволяет сопоставить возвращенный сертификат с терминалом.
- **CertificateResponse**: передает ответ от MCU с полями, установленными в:
  - **terminalLabel**: идентификация удаленного терминала;
  - **certificateResponse**: это фактически цепочка октетов, ASN.1 кодированных из **EncodedReturnSig** в виде:
    - **generalID**: идентификация терминала назначения;
    - **responseRandom**: значение случайного запроса, сформированное MCU;
    - **requestRandom**: воспроизведенный **sRandom**;
    - **certificate**: передает возвращенный сертификат, в котором **type** указывает тип сертификата в виде OID, а **certificate** переносит цифровой сертификат (см. пункт E.12).

E.7 Сквозная аутентификация (Процедура III)

На рисунке E.1 показан сценарий с использованием прокси, разделяющими GK и EP, при этом используются два различных CryptoTokens для посегментной, а также сквозной аутентификации и/или контроля целостности. **CryptoToken** для посегментной аутентификации применяется только к участку маршрута между двумя объектами, и его приходится заново вычислять на каждом другом участке маршрута. С другой стороны, **CryptoToken** для сквозной аутентификации генерируется только однажды передающей конечной точкой и не изменяется промежуточными узлами при транзитной передаче. Промежуточные узлы могут проверять правильность подписей и сертификатов, переданных в сквозных **CryptoToken**, и они должны передавать **CryptoToken** транзитом.

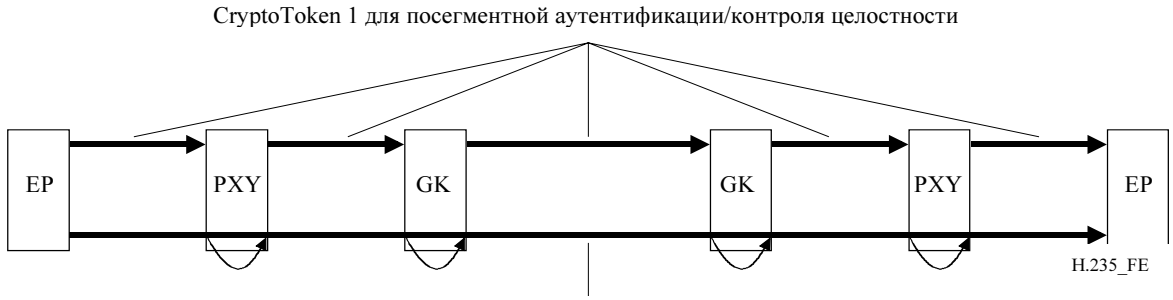


Рисунок E.1/H.235 – Одновременное использование посегментной защиты и сквозной аутентификации

ПРИМЕЧАНИЕ 1. – Прокси может быть отдельным сетевым узлом, как показано на рисунке E.1, или может быть совмещен по функциональным возможностям с таким объектом H.323, как, например, элемент GK.

ПРИМЕЧАНИЕ 2. – В зависимости от сигнального **tokenOID**, прокси способен определить, предназначен ли полученный **CryptoToken** для прокси ("S") или для какого-то другого получателя ("R").

ПРИМЕЧАНИЕ 3. – Вследствие того, что промежуточные объекты изменяют содержимое сигнальных сообщений на каждом участке маршрута, обеспечение сквозной целостности невозможно.

Для фактической сквозной аутентификации через прокси H.323 или через промежуточные элементы сети, передающая конечная точка/терминал должны будут вычислять цифровую подпись следующим образом.

Поле **CryptoH323Token** в каждом сообщении RAS/H.225.0 должно будет содержать следующие поля:

- **nestedCryptoToken**, содержащее **CryptoToken**, которое само содержит **cryptoSignedToken**, включающее следующие поля:
  - **tokenOID**, установленное в:
    - "A", указывающее на то, что вычисление при посегментной аутентификации/контроле целостности учитывает все поля в сообщении RAS/H.225.0 (см. E.9);
    - "B", указывающее на то, что вычисление при аутентификации учитывает только подмножество полей (см. E.8) в сообщении RAS H.225.0 или сообщении о посылке вызова для только-аутентификации.
- **token**, содержащее поля:
  - **toBeSigned**, включающее поле **ClearToken**, используемое со следующими полями:
    - **tokenOID**, установленное в "R", указывает на то, что **ClearToken** используется для только-аутентификации/защиты от неподтверждения[10] при сквозном сценарии;
    - **random** содержит равномерно возрастающий порядковый номер;

[10] Фактическое применение сетевых средств защиты, зависит также от битов использования ключа в сертификате.

- **timestamp**, используемое дополнительно для более совершенной защиты только тогда, когда конечные объекты синхронизированы по времени;
- **generalID** содержит идентификатор конечной точки получателя (только в случае одноадресной передачи). В случае посегментного сценария – это идентификатор следующего сегмента; в случае сквозного сценария – это идентификатор конечной точки на дальнем конце;
- **sendersID** содержит идентификатор конечной точки-отправителя;
- **certificate** содержит цифровой сертификат отправителя, в котором **type** указывает тип сертификата ("V" для сертификатов MD5-RSA или "W" для сертификатов SHA1-RSA), а **certificate** переносит фактический сертификат (см. пункт E.12);
- **dhkey**, используемое для передачи параметров Диффи-Хеллмана во время передачи сообщений **Setup**-to-**Connect**, как это определено в настоящей Рекомендации:
  - **halfkey** содержит случайный открытый ключ одного участника;
  - **modsize** содержит исходный ключ DH (см. таблицу D.4);
  - **generator** содержит DH-группу (см. таблицу D.4).

ПРИМЕЧАНИЕ 4. – Если профиль защиты в виде подписи используется без профиля защиты на основе шифрования речевых сообщений, то не нужно посылать никаких параметров Диффи-Хеллмана, а **dhkey** должен отсутствовать; **halfkey**, **modsize** и **generator** могут быть установлены в {'0'B,'0'B,'0'B}.

- **Token** с полями:
  - **algorithmOID**, установленное в:
    - "V", что указывает на использование подписи MD5-RSA;
    - "W", что указывает на использование подписи SHA1-RSA.
  - **paramS**, установленное равным NULL.
  - **signature**, содержащее подпись, вычисляемую с использованием SHA1-RSA или MD5-RSA по всем полям (если **tokenOID** – "A") или по определенным критическим полям (если **tokenOID** – "B") сообщения RAS H.225.0, или сообщения о посылке вызова.

Прокси может проверить любую полученную цифровую подпись и/или сертификат и может сбросить сообщение, если не считает его соответствующим локальной стратегии, или же прокси должен будет направить полученный **CryptoToken** далее по сети. Прокси должен генерировать новые элементы сигнальной информации H.235 для посегментной защиты согласно Процедурам II или III.

Объект, завершающий участок маршрута, (это может быть терминал), должен будет проверить полученную в **CryptoToken** информацию о защите и, в зависимости от наличия сквозных элементов защиты, он может дополнительно проверить сквозную информацию **CryptoToken**. Точные процедуры проверки в терминале или промежуточном объекте H.323 могут изменяться в соответствии с локальной стратегией.

E.8 Только-аутентификация

Терминалы могут выбирать реализацию такого сетевого средства защиты как "только-аутентификация" (используя OID "B"). В этом случае аутентификатор вычисляется только по подмножеству (**ClearToken** внутри **CryptoToken**) сообщения RAS/H.225.0. Только-аутентификация может быть пригодна для фактической сквозной аутентификации (см. пункт E.7). Следующие поля в структуре **ClearToken** используются как подмножество:

- **tokenOID**: Для реализации только-аутентификации имеется отдельный маркерный идентификатор объекта (**tokenOID** "B").
- **random**: равномерно возрастающий порядковый номер.
- **timeStamp**: временная метка.
- **generalID**: идентификатор получателя (только в случае одноадресных сообщений). В случае посегментного сценария – это идентификатор следующего сегмента; в случае сквозного сценария – это идентификатор конечной точки на дальнем конце.

- **sendersID**: идентификатор отправителя.
- **dhkey**: Параметры Диффи-Хеллмана. Это поле и подполя используются только во время сообщений **Setup**-to-**Connect**.

Аутентификатор вычисляется по **ClearToken** внутри **EncodedGeneralToken** (то есть, **ClearToken**) в **token** структуры **cryptoSignedToken**. Цифровая подпись должна будет вычисляться по закодированной в ASN.1 строке битов в **ClearToken**. Перед вычислением цифровой подписи **tokenOID** в **ClearToken** должен быть установлен в {0 0}.

E.9 Аутентификация и контроль целостности

Для аутентификации и контроля целостности сообщения по всем полям сообщения, кодированного в ASN.1 (с использованием OID "A"), выполняется нижеследующая процедура.

Отправитель сообщения должен будет вычислять подпись следующим образом:

1) Установить значение подписи в конкретную заданную по умолчанию комбинацию фиксированной длины (например, 1024 бита). Этот шаг должен будет зарезервировать пространство для цифровой подписи максимальной длины, которая возможна для данного сертификата. Точная битовая комбинация здесь не имеет значения, но оптимальным вариантом является уникальная битовая комбинация, которая не встречается в оставшейся части сообщения.
2) С помощью ASN.1 кодировать все сообщение; для RAS это должно распространяться на полное сообщение RAS H.225.0; для сообщения о посылке вызова это должно распространяться на полное сообщение о посылке вызова H.225.0.
3) Разместить[11] заданную по умолчанию комбинацию в закодированном сообщении; перезаписать всю установленную битовую комбинацию в виде нулевых битов.
4) Вычислить цифровую подпись по закодированному в ASN.1 сообщению, используя метод, указанный в **algorithmOID** значениями "V" или "W" (см. пункт E.10).
5) Заменить заданную по умолчанию комбинацию в закодированном сообщении вычисленным значением цифровой подписи. В том случае, если цифровая подпись короче, чем зарезервированное пространство, первые нули должны быть поставлены перед наиболее значащими битами значения подписи.

Получатель принимает сообщение и затем обрабатывает его следующим образом:

1) Декодирует сообщение с помощью ASN.1.
2) Извлекает полученное значение цифровой подписи и сохраняет его в локальной переменной SV.
3) Производит поиск и определяет местонахождение значения подписи SV в полученном закодированном сообщении.

ПРИМЕЧАНИЕ. – В тех редких случаях, когда подстрока значения подписи может встречаться в целом сообщении несколько раз, этапы 3–6 должны выполнятся методом последовательных итераций, при этом поиск должен начинаться с различных позиций.

4) Перезаписывает всю битовую комбинацию в закодированном сообщении в виде нулей.
5) Вычисляет цифровую подпись по закодированному сообщению, используя метод, указанный в **algorithmOID** значениями "V" или "W" (см. E.10).
6) Сравнивает SV с вычисленным значением подписи. Сообщение считается не поврежденным и подлинным только в том случае, если оба значения подписи одинаковы; в этом случае аутентификация прошла успешно, и процедура прекращается.
7) В противном случае, повторяет этапы 3–7, восстанавливая SV в его предыдущем местонахождении и производя поиск другого соответствия. Если ни одно из соответствий не дало правильного сравнения значения подписи, то аутентификация считается безуспешной, а сообщение измененным (случайно или преднамеренно) во время пересылки или по какой-то иной причине.

---

[11] Этот этап может включать некоторые эмпирические шаги в том редком случае, когда заданная по умолчанию комбинация встречается в сообщении более одного раза.

E.10 Вычисление цифровой подписи

Исходными данными для процесса формирования цифровой подписи являются – кодированная в ASN.1 битовая строка, а также – результат процесса вычисления дайджеста сообщения и личный ключ подписывающего лица. Организация процесса формирования цифровой подписи зависит от используемого алгоритма подписи; сертификат определяет применяемый алгоритм подписи; если в сертификате присутствует расширение использования ключа, то бит **digitalSignature** должен быть согласован с ключом, чтобы он подходил в качестве подписи. Создаваемое подписывающим лицом значение подписи кодируется как битовая строка и переносится в поле **signature**.

Для вычисления основывающейся на RSA цифровой подписи должен будет использоваться метод, описанный в документе [PKCS #1, раздел E.8.1.1], с приложениями (RSASSA-PKCSl-vl_5-SIGN), процедурами OS2IP, RSASP1, 12OSP и методом кодирования EMSA-PKCS1-v1_5.

E.11 Проверка цифровой подписи

Исходными данными для процесса проверки подписи являются – результат процесса вычисления дайджеста сообщения и открытый ключ подписывающего лица. Получатель может получить правильный открытый ключ для подписывающего лица любым способом, но предпочтительным является его получение из сертификата, который извлечен из поля **certificate** и подтвержден с помощью хеш-величины сертификата подписывающего лица. Подтверждение открытого ключа подписывающего лица может базироваться на выполнении последовательности операций сертификации (RFC 3280). Организация процесса проверки подписи зависит от используемого алгоритма подписи.

Для проверки основывающейся на RSA цифровой подписи должен будет применяться метод, описанный в [PKCS #1, раздел E.8.1.2], с приложением (RSASSA-PKCS1-v1_5-VERIFY), процедурами OS2IP, RSAVP1, I2OSP и методом EMSA-PKCS1-v1_5-ENCODE.

E.12 Манипулирование сертификатами

Для проверки цифровых подписей принимающий объект должен иметь доступ к сертификату отправителя, который подписан официально признанным сертификационным центром (CA). Существуют несколько возможностей касательно способа получения доступа к сертификату отправителя со стороны получателя:

- Сертификат включается в обмен сообщениями, как описано в Процедурах II и III; в этом случае **certificate** содержит фактический сертификат, а **type** содержит OID "V" или OID "W".
- Получатель знает, что сертификат, возможно, сохранен локально от предыдущего обмена.
- Вместо включения самого сертификата, отправитель обеспечивает унифицированный указатель местонахождения ресурса (URL), где может быть найден сертификат. Для этого **certificate** содержит URL, а **type** устанавливается в OID "P".
- Получатель получает сертификат некоторым другим способом, не оговоренным в данной Рекомендации (например, посредством поиска по каталогу LDAP).

В любом случае, когда цифровой сертификат передается в сообщении, принимающий объект (контроллер доступа, конечная точка) должен проверить идентичность отправителя (контроллера доступа, конечной точки) и сертификата с тем, чтобы избежать попытки нарушения защиты со стороны постороннего лица.

Существуют различные возможности для проверки идентичности контроллера доступа при пересылке сообщений с цифровой подписью от контроллера доступа к конечной точке:

– Если имя хоста присутствует, к примеру, в атрибуте общего имени поля subject или поля subjectAltName в сертификате, то конечная точка может проверить это имя хоста, сравнив его с идентификатором контроллера доступа. Кроме того, конечная точка может использовать DNS для запроса соответствующего IP-адреса и сравнить его с IP-адресом контроллера доступа, представленным в сообщении ответа, имеющем подпись контроллера доступа.

– К примеру, идентификатор контроллера доступа может быть составлен из IP-адреса (представленного в виде 4-байтовой величины в порядке следования байтов в сети), сцепленного с другими идентифицирующими данными идентификатора контроллера доступа и усеченного до максимальной длины поля ID отправителей, которое переносит идентификатор контроллера доступа. Конечная точка может дополнительно проверить IP-адрес, относящийся к имени хоста, и

сравнить его с IP-адресом, представленным в заголовке IP ответного сообщения контроллера доступа.

ПРИМЕЧАНИЕ. – Этот метод не срабатывает при задействии устройств трансляции сетевых адресов (NAT).

– Если имя хоста отсутствует в сертификате, то IP-адрес, который должен входить в состав сертификата (*iPAddress subjectAltName*), должен быть непосредственно взят для выполнения вышеуказанных проверок.

Пользователи должны тщательно исследовать сертификат, представленный контроллером доступа, с тем, чтобы определить соответствие его их предположениям. Если конечная точка имеет дополнительную информацию о предполагаемой идентичности контроллера доступа, то имя хоста может быть пропущено. К примеру, конечная точка может подключаться к контроллеру доступа, адрес и имя хоста которого являются динамическими, но конечная точка знает сертификат, который представит контроллер доступа. В таких случаях важно, по возможности, сократить область применения соответствующих сертификатов с тем, чтобы предотвратить попытки нарушения защиты со стороны посторонних лиц. В особых случаях для конечной точки, вероятно, лучше просто проигнорировать идентичность контроллера доступа, но при этом необходимо осознавать, что это сделает соединение открытым для активных попыток нарушения защиты.

Если имя хоста не соответствует его идентификатору в сертификате, то ориентированные на пользователя конечные точки должны или уведомить пользователя (конечные точки могут дать пользователю возможность, в любом случае, продолжить соединение), или завершить соединение при грубой ошибке в сертификате. Автоматически управляемые конечные точки должны зарегистрировать ошибку в соответствующем журнале регистрации сетевых событий (если он имеется) и должны завершить соединение (при грубой ошибке в сертификате).

Автоматически управляемые конечные точки могут обеспечить установку такой конфигурации, которая неспособна на подобную проверку, но они должны обеспечивать установку, позволяющую производить эту проверку.

Аналогично, рекомендуется, чтобы контроллер доступа выполнял проверку на идентичность для любых сообщений с цифровой подписью, посланных от конечной точки к контроллеру доступа. Конкретный способ введения такой проверки – это вопрос местной реализации, который должен зависеть от реализации стратегии защиты контроллера доступа. К примеру, можно вообразить, что имя пользователя, передаваемое в сертификате, также может входить в состав идентификатора H.323. Более того, контроллер доступа может организовать перекрестную сверку таких идентифицирующих данных с локально администрируемыми/конфигурируемыми данными пользователя, если они имеются, и может основывать на них свои решения в части стратегии.

Если контроллер доступа имеет дополнительную информацию о предполагаемой идентичности конечной точки, то проверка имени хоста может быть опущена. К примеру, контроллер доступа может подключаться к конечной точке, адрес и имя хоста которой являются динамическими, но контроллер доступа осведомлен о сертификате, который представит конечная точка. В таких случаях важно, по возможности, сократить область применения соответствующих сертификатов с тем, чтобы предупредить попытки нарушения защиты со стороны посторонних лиц. В особых случаях, контроллеру доступа лучше просто проигнорировать идентичность конечной точки, но при этом необходимо осознавать, что это сделает соединение открытым для активных попыток нарушения защиты.

Если имя хоста не соответствует идентификатору в сертификате, то контроллер доступа должен будет зарегистрировать ошибку в соответствующем журнале регистрации сетевых событий (если он имеется) и должен будет завершить соединение (при грубой ошибке в сертификате).

Если присутствует расширение subjectAltName типа dNSName, то оно должно использоваться в качестве идентификатора. В ином случае, должно использоваться поле (более конкретного) Common Name в поле Subject сертификата. Хотя на практике используется Common Name, против него имеются возражения и, сертификационным центрам рекомендовано использовать вместо этого dNSName.

Согласование должно производиться с использованием правил согласования, определенных в RFC 3280. Если в сертификате присутствуют несколько идентификаторов (к примеру, несколько имен dNSName), то согласование любого из них считается приемлемым. Имена могут содержать кодовую комбинацию для универсального сопоставления (wildcard)*, которая считается соответствующей любому отдельному компоненту имени домена или фрагменту компонента. К примеру, *.a.com соответствует foo.a.com, но не bar.foo.a.com. f*.com соответствует foo.com, но не bar.com.

Процедуры II и III обеспечивают средства для передачи цифрового сертификата. Для повышения эффективности цифровые сертификаты объектов должны передаваться в большинстве случаев однократно, если эти объекты уже не добыли их другим, не оговоренным настоящей Рекомендацией способом. Обмен сертификатами должен происходить только в начале процесса установления связи: для сообщения RAS это происходит или во время выявления контроллера доступа или, если эта фаза опущена, то во время регистрации контроллера доступа. То же самое относится и к быстрому соединению, при этом сертификат может быть включен в начальные сообщения о посылке вызова, но его допустимо опускать в более поздних сообщениях о посылке вызова.

Для этого профиля защиты должен будет использоваться сертификат X.509v3 (1997 г.). Другие форматы сертификатов являются предметом дальнейшего изучения.

E.13 Пример использования Процедуры II

Рассмотрим случай на рисунке E.2, где каждый объект имеет свою собственную пару личных открытых ключей/сертификат. Объект также может обладать множеством пар ключей. На рисунке прокси H.323 отделяет EP1 от GK1.

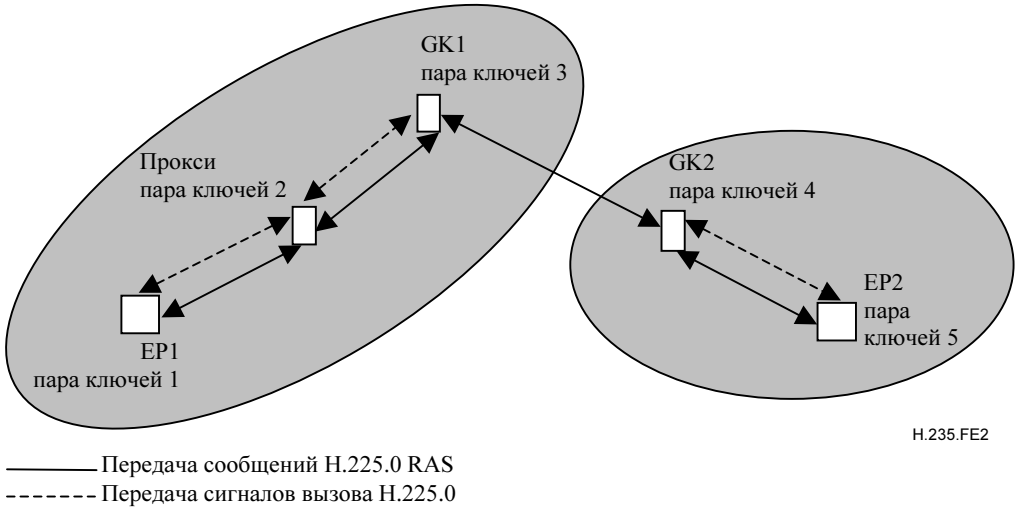


Рисунок E.2 /H.235 – Пример использования открытого ключа в модели маршрутизации GK-GK.

Прокси H.323 действует в двойном режиме. С одной стороны, прокси завершает аутентификацию и контроль целостности на каждом из своих участков маршрута. Прокси активно включает последнюю вычисленную информацию по аутентификации/контролю целостности в исходящие сообщения RAS способом, описанным в Процедуре I Приложения D. С другой стороны, прокси обеспечивает передачу сквозной информации о защите без ее модификации. Прокси может, кроме того, проверять полученные сертификаты и/или цифровые подписи при транзите.

Ниже подробно показаны процедуры аутентификации, контроля целостности и защиты от неподтверждения для сообщений RAS, сообщений о посылке вызова H.225.0 и сообщений H.245.

E.13.1 Аутентификация, контроль целостности сообщений RAS и защита от неподтверждения

Рассмотрим случай посегментной связи, когда EP1 желает послать сообщение RAS, к примеру, сообщение **ARQ** к GK1. EP1 генерирует временную метку и порядковый номер и включает их в поля **timeStamp** и **random**, соответственно, наряду с псевдонимом прокси в поле **generalID** и **sendersID** конечной точки EP1. Эти поля присутствуют в поле **ClearToken** представления **EncodedGeneralTokens**, присутствующего в **token** поля **cryptoSignedToken** в поле **CryptoToken** маркера **cryptoH323Token** сообщения **ARQ**. Этот **cryptoH323Token** является одним из, по крайней мере, нескольких маркеров в последовательности **cryptoTokens**. **tokenOID** в **cryptoSignedToken** устанавливается в "A" и указывает на то, что все поля в сообщении **ARQ** подписаны. **token** в **cryptoSignedToken** содержит **algorithmOID**, установленный в "V", что указывает на использование MD5-RSA, или же **algorithmOID**, установленный в "W" и указывающий на использование SHA1-RSA, и **params**, установленный в NULL. Затем EP1

вычисляет подпись, основываясь на данном алгоритме подписи и используя свой личный ключ. Подпись вычисляется по всем полям сообщения **ARQ**, если **tokenOID** имеет значение "A". EP1 включает вычисленную подпись в рамках **signature** в поле **token** поля **cryptoSignedToken** в представлении **CryptoToken** в маркере **cryptoH323Token** сообщения **ARQ** и включает свой сертификат в поле **certificate**.

Аналогично, для сквозной связи между конечными точками через прокси EP1 формирует другой **CryptoToken**, содержащий цифровую подпись, которая охватывает некоторые критические поля (см. E.7) в **ClearToken** сообщения **ARQ**. **tokenOID** в **CryptoSignedToken** устанавливается в "B", что указывает на только-аутентификацию этого **ClearToken**; установки **tokenOID** в **ClearToken** в "R", указывают на сквозную аутентификацию, кроме того, для **timeStamp**, **random**, **sendersID**, **generalID** (и **dhkey** в случае **SETUP/CONNECT**) в **token** устанавливаются следующие поля: **algorithmOID** устанавливается в "V" или "W", что указывает на алгоритм формирования подписи, **paramS** устанавливается в NULL, а **signature** – в вычисленные по полям **ClearToken** значения цифровой подписи. Поле **certificate** переносит цифровой сертификат EP1. Затем сообщение **ARQ** посылается к прокси.

По получении сообщения **ARQ**, прокси проверяет подпись тех маркеров, которые ему адресованы (в данном случае, например, таких, у которых **tokenOID** установлен в "A"). Прокси осуществляет проверку, основываясь на нескольких критериях, которые включают:

- жизнеспособность **timestamp**, уникальность **random**;
- идентичность **generalID** и собственного идентификатора;
- разрешения на доступ для **sendersID**;
- соответствие подписи в сообщении **ARQ** и подписи, вычисленной посредством GK1;
- проверку параметров Диффи-Хеллмана (например, проверяется правильность 1024-битового исходного ключа и generator) защищенности параметров DH является процессом, требующим много времени, и может быть осуществлена только тогда, когда этого требует локальная стратегия;
- проверку полученного сертификата.

Если подпись успешно проверена, то прокси вычисляет новую подпись, чтобы вставить ее (заменить) в сообщение **ARQ** перед его отправкой GK1, следующим образом. Прокси заменяет поля **timeStamp**, **random**, **sendersID** и **generalID** в поле **ClearToken** (**toBeSigned**), используя значения, соответствующие участку маршрута прокси – GK1. Поле **timeStamp** содержит текущую временную метку, поле **random** содержит следующий равномерно возрастающий порядковый номер для участка маршрута прокси – GK1, **sendersID** прокси, а поле **generalID** – псевдоним GK1. Затем прокси вычисляет новую подпись для этого сообщения **ARQ**, используя свой личный ключ и алгоритм подписи, вставляет ее в поле **signature** в **token** и добавляет свой **certificate**. Прокси также включает полученный сквозной **CryptoToken** с его **ClearToken** в новое исходящее сообщение и передает сообщение **ARQ** к GK1. Вычисленная посредством EP1 на основе выбранных полей сообщения **ARQ** подпись (значение **tokenOID** равно "B"), которая не предназначается для прокси, также передается в нетронутом виде в сообщении **ARQ** к GK1.

По получении сообщения **ARQ**, GK1 проверяет подписи, вычисляет новую подпись после соответствующего изменения полей **ClearToken** в **toBeSigned**, вставляет ее в поле **signature**, добавляет свой сертификат и передает сообщение **Setup** к EP2. И снова GK1 должен переслать любую сквозную информацию, полученную в отдельном **CryptoTokens**, к одноранговому GK2, включая эту информацию в отдельный **CryptoToken** без изменений.

E.13.2 Только-аутентификация сообщений RAS

Рассмотрим случай посегментной связи, когда EP1 желает послать сообщение RAS, к примеру, сообщение **ARQ** к GK1. EP1 генерирует временную метку и порядковый номер и включает их в поля **timeStamp** и **random**, соответственно, вместе с псевдонимом прокси в поле **generalID** и идентификатором EP в **sendersID**. Эти поля присутствуют в поле **ClearToken** представления **toBeSigned**, содержащегося в **token** в **cryptoSignedToken** поля **CryptoToken** маркера **cryptoH323Token** сообщения **ARQ**. **tokenOID** в **cryptoSignedToken** устанавливается в "B", указывающее, что подписано только конкретное подмножество полей в **ClearToken**. **token** в **cryptoSignedToken** имеет **algorithmOID**, установленный в "V", что указывает на использование MD5-RSA, или он установлен в "W", что указывает на применение алгоритма подписи SHA1-RSA, и **paramS**, установленный в NULL. Затем EP1 вычисляет подпись на основе алгоритма подписи, используя свой личный ключ. Подпись вычисляется по

конкретным полям **ClearToken** сообщения **ARQ**. EP1 включает вычисленную подпись в рамках **signature** в поле **token** поля **cryptoSignedToken** представления **CryptoToken** в маркере **cryptoH323Token** сообщения **ARQ** и добавляет свой **certificate**.

Аналогично, EP1 формирует другую цифровую подпись для сквозной аутентификации, которая охватывает некоторые поля **ClearToken** в отдельном **CryptoToken** в сообщении **ARQ**. Эта цифровая подпись (идентифицируемая посредством значений **tokenOID**, равных "V" или "W") включается. Затем сообщение **ARQ** посылается к прокси.

По получении сообщения **ARQ,** прокси проверяет подпись тех маркеров, которые адресованы ему (в данном случае, например, таких, у которых **tokenOID** установлен в "B"). Данная проверка основывается на нескольких критериях, включая:

- жизнеспособность **timestamp**, уникальность **random**;
- идентичность **generalID** и собственного идентификатора;
- разрешения на доступ для **sendersID**;
- соответствие подписи в сообщении **ARQ** и подписи, вычисленной GK1;
- проверку полученного сертификата.

Если подпись успешно проверена, то прокси вычисляет новую подпись, чтобы вставить ее (заменить) в сообщение **ARQ** перед отправкой его к GK1, следующим образом. Прокси заменяет поля **timeStamp**, **random**, **sendersID** и **generalID** в поле **ClearToken** маркера **toBeSigned**, используя значения, соответствующие участку маршрута прокси – GK1. Поле **timeStamp** содержит текущую временную метку, поле **random** содержит следующий равномерно возрастающий порядковый номер для участка маршрута прокси – GK1, а поле **generalID** содержит псевдоним GK1. Затем прокси вычисляет новую подпись для этого **ClearToken**, используя свой личный ключ и алгоритм подписи MD5-RSA или SHA1-RSA (**algorithmOID** установлен в "V" или "W"), вставляет ее в **signature** в маркере **cryptoSignedToken**, добавляет свой **certificate** и передает сообщение **ARQ** к GK1. Подпись, вычисленная EP1 на основе выбранных полей **ClearToken** сообщения **ARQ** (**tokenOID** установлен в "B"), которая не предназначалась для прокси, также передается нетронутой в сообщении **ARQ** к GK1.

По получении сообщения **ARQ,** GK1 проверяет подпись, вычисляет новую подпись после должного изменения соответствующих полей **ClearToken** в маркере **toBeSigned**, вставляет ее в поле **signature** и передает сообщение **Setup** к EP2. Информация о сквозной подписи от EP1 включается в сообщение **Setup** в нетронутом виде.

E.13.3 Аутентификация, контроль целостности и защита от неподтверждения сообщений H.225.0

Процедура для сообщений H.225.0 идентична процедуре для сообщений RAS**.** Единственное различие состоит в том, что набор полей, которые необходимо подписать, должен быть определен для каждого сообщения о посылке вызова H.225.0, если **tokenOID** установлен в "B".

E.13.4 Аутентификация и контроль целостности сообщений H.245

Рассмотрим случай, когда EP1 желает послать сообщение H.245, к примеру, сообщение **TerminalCapabilitySet** к EP2. EP1 проверяет, должно ли быть сообщение H.225.0 послано к прокси. Если да, то сообщение H.245 туннелируется в рамках сообщения H.225.0. Поля в сообщении H.225.0 устанавливаются, как описано ранее для передачи сообщения H.225.0. Так как сообщение H.245 туннелируется, то **h323-un-pdu** в сообщении **H323-UserInformation** содержит поля со следующими установленными значениями:

- поле **h323-message-body** устанавливается в значение типа передаваемого сообщения H.225.0;

- поле **h245Tunnelling** устанавливается в значение TRUE;
- поле **h245Control** содержит последовательность октетов PDU H.245.

Однако, если передача сообщения H.225.0 не отложена, то сообщение H.245 туннелируется в рамках специального сообщения **facility** H.225.0. Поле **h323-uu-pdu** в сообщении **h323-UserInformation** содержит следующие установки полей:

- поле **h323-message-body** устанавливается в значение **facility**, при этом:
  - **reason** устанавливается в **undefinedReason**;
  - **tokens** и **cryptoTokens** устанавливаются как для любого сообщения H.225.0.
- **h245Tunnelling** устанавливается в значение TRUE.
- **h245Control** содержит последовательность октетов PDU H.245.

Затем сообщение **facility** передается посредством EP1 к прокси.

В любом случае (если передача сообщения H.225.0 отложена или используется специальное сообщение **facility** H.225.0), то, по получении этого сообщения, прокси проверяет подпись, которая предназначена для него (в таком случае она показана значением **tokenOID**, равным "A"). Затем, если передача сообщения H.225.0 отложена на участке маршрута прокси – GK1, то сообщение H.245 туннелируется в рамках этого сообщения; в ином случае оно туннелируется в рамках специального сообщения **facility** H.225.0. Как и в случае передачи любого сообщения о посылке вызова H.225.0, для сообщения H.225.0 вычисляется новая подпись, до его передачи от прокси к GK1. Подпись, которая была послана от EP1 к прокси и не предназначалась для прокси, передается прокси к GK1 нетронутой.

В настоящем пункте приведено краткое описание способа и средств обеспечения защиты посредством профиля защиты в виде подписи различных сигнальных сообщений H.323.

E.14 Совместимость с H.235 версии 1

При том, что эти профили защиты разработаны с учетом H.235, версии 2 [H.235v2], также возможно применение этих профилей защиты для H.235 версии 1 [H.235v1] с некоторыми небольшими изменениями. Получатель способен обнаружить присутствие версии протокола H.235 отправителя, путем определения идентификаторов объектов профиля защиты (см. E.18).

Ограничения H.235 версии 1 [H.235v1]:

- не устанавливать или не определять **sendersID** в **ClearToken**.

E.15 Режим многоадресной передачи

Многоадресные сообщения H.225.0, такие как **GRQ** или **LRQ**, должны будут включать **CryptoToken** согласно Процедурам II и III, где значение **generalID** не установлено. Если такие сообщения – одноадресные, то эти сообщения должны будут включать **CryptoToken**.

E.16 Список защищенных сигнальных сообщений

E.16.1 Сообщение RAS H.225.0

Сообщение RAS H.225.0	Сигнальные поля H.235	Только-аутентификация	Аутентификация и контроль целостности	Защита от неподтверждения
Любое	cryptoTokens	Процедура II/III	Процедура II/III	Процедура II/III

ПРИМЕЧАНИЕ. – Для одноадресных сообщений Процедуры II или III будут применяться с полями защиты в используемом **CryptoToken**.

E.16.2 Передача сигналов вызова H.225.0

Сообщения о посылке вызова H.225.0	Сигнальные поля H.235	Только-аутентификация	Аутентификация и контроль целостности	Защита от неподтверждения
Alerting-UUIE, CallProceeding-UUIE, Connect-UUIE, Setup-UUIE, Facility-UUIE, Progress-UUIE, Information-UUIE, ReleaseComplete-UUIE, Status-UUIE, StatusInquiry-UUIE, SetupAcknowledge-UUIE, Notify-UUIE	cryptoTokens	Процедура II/III	Процедура II/III	Процедура II/III

E.17 Использование sendersID и generalID

ClearToken содержит поля **sendersID** и **generalID**. При наличии идентифицирующей информации, и **sendersID** должен быть установлен в значение идентификатора контроллера доступа (GKID) для инициированного контроллером доступа сообщения и – в значение идентификатора конечной точки (EPID) для инициированных конечной точкой сообщений. При наличии идентифицирующей информации, **generalID** должен быть установлен в значение GKID для инициированных конечной точкой сообщений и – в значение EPID для сообщений, инициированных контроллером доступа. При отсутствии идентифицирующей информации отсутствует или в случае, когда результат многоадресной/широковещательной передачи неоднозначен, это поле пропускается или оно должно будет содержать нулевую строку. В таблице E.2 приведены все описанные случаи.

Таблица E.2/H.235 – Идентификаторы объектов, используемые в Приложении E

Сообщение	sendersID	generalID
Одноадресное **GRQ**	**EPID**, если имеется, в ином случае – **NULL**	**GKID**
Многоадресное **GRQ**	**EPID**, если имеется, в ином случае – **NULL**	
GCF, **GRJ**	**GKID**	**EPID** если имеется, в ином случае – **NULL**
Исходное **RRQ**		**GKID**
RCF	**GKID**	**EPID**
RRJ	**GKID**	
URQ, **UCF**, **URJ**, **BRQ**, **BCF**, **BRJ**, **DRQ**, **DCF**, **DRJ**, **NSM**, **RIP**, **SCI**, **SCR**, **XRS** (от EP к GK)	**EPID**	**GKID**
URQ, **UCF**, **URJ**, **BRQ**, **BCF**, **BRJ**, **DRQ**, **DCF**, **DRJ**, **NSM**, **RIP**, **SCI**, **SCR**, **XRS** (от GK к EP)	**GKID**	**EPID**
ARQ, **IRQ**, **RAI**	**EPID**	**GKID**
ACF, **ARJ**, **BCF**, **LCF**, **LRJ**, **IRR**, **IRQ**, **RAC**, **LCF**, **LRJ**, **IACK**, **INAK**	**GKID**	**EPID**
Одноадресное **LRQ** (от EP к GK)	**EPID**	**GKID**
Одноадресное **LRQ** (от GK к GK)	**GKID**	**GKID**
Многоадресное **LRQ**	**EPID**	
ПРИМЕЧАНИЕ. – GKID обозначает идентификатор контроллера доступа, EPID обозначает идентификатор конечной точки. Пробел указывает на отсутствие или нулевую идентифицирующую строку.		

E.18 Перечень идентификаторов объектов

В таблице E.3 перечислены все идентификаторы объектов, на которые делаются ссылки (см. также [OIW] и [WEBOID]). Здесь также перечислены идентификаторы объектов для H.235vl [H.235v1] и H.235v2 [H.235v2]

Таблица E.3/H.235 – Идентификаторы объектов, используемые Приложением E

Опорное значение идентификатора объекта	Значение(я) идентификаторов объектов	Описание
"A"	{itu-t (0) recommendation (0) h (8) 235 version (0) 2 1} {itu-t (0) recommendation (0) h (8) 235 version (0) 1 1}	Используется в Процедуре II для CryptoToken – tokenOID и указывает на то, что подпись включает **все** поля в сообщении RAS/H.225.0 или сообщении о посылке вызова (аутентификация и контроль целостности).
"B"	{itu-t (0) recommendation (0) h (8) 235 version (0) 3 2} {itu-t (0) recommendation (0) h (8) 235 version (0) 2 2} {itu-t (0) recommendation (0) h (8) 235 version (0) 1 2}	Используется в Процедуре II для CryptoToken – tokenOID и указывает на то, что подпись включает **подмножество** полей в сообщении RAS/H.225.0 (ClearToken) для терминалов, осуществляющих только-аутентификацию без контроля целостности. Используется в Процедуре IA Приложения D для CryptoToken-tokenOID и указывает на то, что хеш-величина включает подмножество полей в сообщении RAS/H.225.0 (ClearToken) для терминалов с только-аутентификацией, но без контроля целостности.
"P"	{itu-t (0) recommendation (0) h (8) 235 version (0) 2 4} {itu-t (0) recommendation (0) h (8) 235 version (0) 1 4}	Используется в Процедурах II или III и указывает на то, что **certificate** передает URL.
"R"	{itu-t (0) recommendation (0) h (8) 235 version (0) 2 3} {itu-t (0) recommendation (0) h (8) 235 version (0) 1 3}	Используется в Процедуре II для CryptoToken – tokenOID и указывает на то, что ClearToken используется для сквозной аутентификации и контроля целостности.
"S"	{itu-t (0) recommendation (0) h (8) 235 version (0) 2 7} {itu-t (0) recommendation (0) h (8) 235 version (0) 1 7}	Используемый в Процедуре II этот маркер OID указывает на аутентификацию сообщения, контроль целостности и защиту от неподтверждения.
"V"	{iso(1) member-body(2) us(840) rsadsi(113549) pkcs(1) pkcs-1(1) 4}	Используется в Процедуре II или в Процедуре III как алгоритм OID, указывающий на использование цифровой подписи MD5-RSA.
"W"	{iso(1) member-body(2) us(840) rsadsi(113549) pkcs(1) pkcs-1(1) 5}	Используется в Процедуре II или в Процедуре III как алгоритм OID, указывающий на использование цифровой подписи SHA1-RSA.

Приложение F

Смешанный профиль защиты

Резюме

Цель данного приложения – описание эффективного и расширяемого, основывающегося на PKI, смешанного профиля защиты для Рекомендации МСЭ-Т H.235, версии 2. Этот смешанный профиль защиты, представленный в данном Приложении, опирается на профили защиты из Приложения D и Приложения E путем использования цифровых подписей из Приложения E и базового профиля защиты из Приложения D.

F.1 Общее описание

В данном Приложении описывается эффективный и расширяемый, основывающийся на PKI, смешанный профиль защиты, использующий цифровые подписи из Приложения E и базовый профиль защиты из Приложения D. Данное Приложение предлагается в качестве опции. Объекты защиты H.323 (терминалы, контроллеры доступа, шлюзы, MCU и т. д.) могут реализовывать этот смешанный профиль защиты для ее усовершенствования или для других целей.

Понятие "смешанный" в данном тексте должно означать, что процедуры защиты из профиля защиты в виде подписи в Приложении E фактически применяются не в полную силу, а процедуры формирования цифровых подписей все же соответствуют процедурам RAS. Кроме того, цифровые подписи применяются только при абсолютной необходимости, в ином случае используются высокоэффективные симметричные методы защиты из базового профиля защиты в Приложении D.

Смешанный профиль защиты применим для масштабируемой "глобальной" IP-телефонии. Этот профиль защиты не имеет ограничений простого, базового профиля защиты из Приложения D, при его непосредственном применении. Кроме того, этот профиль защиты не имеет некоторых недостатков профиля защиты из Приложения E, таких как потребность в более широкой полосе частот и повышенные требования к качеству обработки, при его непосредственном применении. К примеру, смешанный профиль защиты не зависит от (статического) административного управления общими "ключами" сетевых сегментов в различных доменах. Таким образом, пользователям легче выбрать поставщика VoIP. Так что этот профиль защиты обеспечивает в некотором роде и мобильность пользователя. Он использует асимметричную криптографию с подписями и сертификатами только в случае необходимости, в ином случае используются более простые и более эффективные симметричные методы. Он обеспечивает туннелирование сообщений H.245 для контроля целостности сообщений H.245, а также реализует некоторые условия для защиты от неподтверждения сообщений.

Этот смешанный профиль защиты предусматривает модель с маршрутизацией посредством GK и базируется на методах туннелирования H.245. Вопрос обеспечения моделей, имеющих другой тип маршрутизации, подлежит дальнейшему изучению.

Возможности, предоставляемые этим профилем, включают:

Для сообщений RAS H.225.0 и H.245:

- Аутентификацию пользователя требуемым объектом независимо от количества сетевых сегментов[12] прикладного уровня, которые проходит сообщение.

- Контроль целостности всех критических частей (полей) сообщения, поступающих к объекту, независимо от количества сетевых сегментов прикладного уровня, которые проходит сообщение. Контроль целостности самого сообщения с использованием четко сформированного случайного числа также является необязательным.

---

[12] Под сетевым сегментом здесь понимается доверительный сетевой элемент H.235 (к примеру, GK, GW, MCU, прокси или брандмауэр). Таким образом, посегментная защита прикладного уровня при использовании симметричных методов не обеспечивает истинную сквозную защиту между терминалами.

- Посегментную аутентификацию сообщений прикладного уровня, контроль целостности и (до некоторой степени) защиту от неподтверждения, причем эти сетевые средства защиты обеспечиваются для полного сообщения.
- Используя имеющуюся инфраструктуру открытых ключей, пользователи могут выбирать своего поставщика услуг. Процедура управления ключами, в данном случае – сеансовыми, органично интегрируется в этот смешанный профиль защиты.

Соответствующее обеспечение описанных выше сетевых средств защиты будет препятствовать некоторым типам попыток нарушения защиты, включая:

- *Попытки нарушения защиты со стороны посторонних лиц:* Посегментная аутентификация сообщений прикладного уровня и контроль целостности препятствуют таким попыткам, когда это постороннее лицо находится в сетевом сегменте прикладного уровня, к примеру, "недружелюбный" маршрутизатор.
- *Повторные попытки нарушения защиты:* Использование временных меток и порядковых номеров предупреждает такие попытки.
- *Спуфинг (имитация соединения):* Аутентификация пользователя предупреждает такие попытки.
- *"Захват" соединения:* Использование аутентификации/контроля целостности для каждого сигнального сообщения предупреждает такие попытки.

F.2 Нормативные источники ссылок

Следующие Рекомендации МСЭ-Т и другие источники содержат положения, которые, будучи упомянутыми, в качестве ссылок в данном тексте, составляют положения данной Рекомендации. На момент публикации указанные издания были действующими. Все Рекомендации и другие источники ссылок подлежат пересмотру; поэтому всем пользователям этих Рекомендаций предлагается рассмотреть возможность использования самого последнего издания этих Рекомендаций и других источников ссылок, перечисленных ниже. Перечень действующих рекомендаций МСЭ-Т публикуется регулярно. Ссылка на любой документ в рамках данной Рекомендации не придает ему, даже при том, что это отдельный документ, статуса Рекомендации.

– Рекомендация МСЭ-Т H.225.0, версия 4 (2000 г.), *Протоколы передачи сигналов вызова и пакетирование мультимедийных потоков для систем мультимедийной связи в пакетном режиме.*

– Рекомендация МСЭ-Т H.235.0, версия 2 (2000 г.), *Средства защиты и шифрования информации для мультимедийных терминалов серии H (терминалов H.323 и других, основывающихся на H.245).*

– Рекомендация МСЭ-Т H.245.0, версия 8 (2001 г.), *Протокол управления для мультимедийной связи.*

– Рекомендация МСЭ-Т H.323, версия 4 (2000 г.), *Системы мультимедийной связи в пакетном режиме.*

– Стандарт IETF 3280 (2002 г.), *Сертификат на инфраструктуру открытого ключа Интернет X.509 и профиль Перечня Отмененных Сертификатов (CRL).*

F.3 Акронимы

В этом Приложении определены следующие акронимы:

GCF Квитирование контроллера доступа

GK Контроллер доступа

GRQ Запрос контроллера доступа

ICV Контрольный признак целостности

LRQ Запрос местонахождения

OID Идентификатор объекта

RAS Регистрация, доступ и статус

RCF Подтверждение регистрации

RRQ Запрос регистрации

RSA Алгоритм шифрования с открытым ключом Райвеста-Шамира-Адлемана

SHA Алгоритм аутентификации и проверки целостности информации

URQ Запрос на отсутствие регистрации

F.4 Принятые термины и условные обозначения

Смешанный профиль защиты использует термины и определения из Приложений D и E.

При том, что сетевое средство защиты в виде контроля целостности сообщений всегда обеспечивает и аутентификацию сообщений, обратное утверждение не всегда верно. При режиме только-аутентификации гарантированный контроль целостности охватывает только определенное подмножество полей сообщения. Это касается сетевых средств контроля целостности, реализуемых асимметричными методами (к примеру, цифровые подписи). Таким образом, на практике комбинированное сетевое средство защиты в виде аутентификации и контроля целостности использует те же самые данные ключа, не ослабляя при этом защиту.

Этот профиль защиты применим в условиях потенциального наличия множества терминалов, когда невозможно осуществить присвоение статического пароля/симметричного ключа, к примеру, при широкомасштабных или глобальных сценариях. Вместо этого, этот профиль защиты предполагает наличие инфраструктуры открытого ключа с присвоенными сертификатами и каталогами личных/открытых ключей и т. д. Кроме того, этот профиль защиты использует там, где это приемлемо, методы симметричного шифрования.

Этот профиль защиты вводит термины – "первое" посланное сообщение и "последнее" посланное сообщение. Обеспечение секретности первого сообщения (и, вероятно, также и последнего) отличается от обеспечения секретности остальных сообщений другого типа.

Под "первым" посланным сообщением подразумевается сообщение, которое передается потоком между двумя объектами H.323 и создает контекст для обеспечения секретности. Оно дает доступ к данным симметричного ключа обоих объектов и, к примеру, маркирует начало вызова. Для сообщений RAS H.225.0 первым сообщением является RRQ и соответствующее сообщение ответа. Для сообщений о посылке вызова H.225.0 с использованием быстрого старта первым сообщением является SETUP и CONNECT.

"Последнее" сообщение завершает созданный контекст для обеспечения секретности. Сформированные данные ключа должны быть уничтожены. Для сообщений RAS H.225.0 последним сообщением является URQ и соответствующее сообщение ответа, тогда как для сообщений о посылке вызова H.225.0 последним сообщением является RELEASE-COMPLETE.

Этот профиль защиты предполагает модель маршрутизации и с помощью GK с применением метода передачи сигналов вызова при быстром соединении. Сообщения управления вызовом H.245 безопасно туннелируются в сообщениях о посылке вызова H.225.0 и перенимают, таким образом, схему обеспечения секретности H.225.0.

Профиль защиты в виде подписи позволяет туннелировать под защитой – PDU управления вызовом H.245 в рамках сообщений H.225.0. Обновление ключей H.245 и механизмы синхронизации требуют туннелирования для осуществления передачи сообщения обновления ключей FACILITY, также оно пригодно, к примеру, для очень продолжительных вызовов.

Заштрихованные по диагонали позиции в таблице F.1 представляют те механизмы защиты, которые используются смешанным профилем защиты.

ПРИМЕЧАНИЕ. – Сертификаты RSA с хешированием MD5 не входят в состав этого профиля защиты.

Профиль защиты с шифрованием речевых сообщений из Приложения D (см. пункт D.7) может дополнительно использоваться наряду со смешанным профилем защиты. Его использование согласовано в рамках передачи сигналов об установлении соединения.

Таблица F.1/H.235 – Общее описание смешанного профиля защиты

Сетевые средства защиты	Функции обработки вызова			
	RAS	**H.225.0**	**H.245 (Примечание 3)**	**RTP**
Аутентификация	Цифровая подпись RAS (SHA1)	Цифровая подпись RAS (SHA1)	Цифровая подпись RAS (SHA1)	
	HMAC-SHA1-96	HMAC-SHA1-96	HMAC-SHA1-96	
Защита от неподтверждения	(возможна только в первом сообщении)	(возможна только в первом сообщении)		
Контроль целостности	Цифровая подпись RAS (SHA1)	Цифровая подпись RAS (SHA1)	Цифровая подпись RAS (SHA1)	
	HMAC-SHA1-96	HMAC-SHA1-96	HMAC-SHA1-96	
Конфиденциальность				
Контроль доступа				
Управление ключами	распределение сертификатов	распределение сертификатов		
	Аутентифицированный обмен ключами Диффи-Хеллмана	Аутентифицированный обмен ключами Диффи-Хеллмана		

ПРИМЕЧАНИЕ 1. – Смешанный профиль защиты должен также поддерживаться другими объектами H.235 (к примеру, контроллерами доступа, шлюзами и прокси H.235).

ПРИМЕЧАНИЕ 2. – Имеющиеся в сертификате биты использования ключа также должны определять сетевое средство защиты, обеспечиваемое терминалом (к примеру, заявленная защита от неподтверждения).

ПРИМЕЧАНИЕ 3. – Туннелированное сообщение H.245 и встроенное в H.245 внутри сообщения о быстром соединении H.225.0.

В этом приложении можно применять такое сетевое средство – как обеспечение целостности сообщения, которое охватывает полное сообщение. Для сообщений RAS H.225.0 обеспечение целостности охватывает полное сообщение RAS; для сообщений о посылке вызова это сетевое средство защиты охватывает полное сообщение о посылке вызова H.225.0, включая заголовки Q.931.

Для аутентификации пользователь должен использовать систему подписи с открытым/личным ключами. Такая система обычно обеспечивает лучший контроль целостности.

Данная Рекомендация не описывает процедуры регистрации, сертификации и распределения сертификатов от доверительного центра, присвоения личных/открытых ключей, службы каталогов, специальные параметры CA, процедуры отмены сертификатов, восстановления/обновления пар ключей и другие процедуры эксплуатации и управления сертификатами, такие как доставка сертификатов или открытых/личных ключей и их установка в терминалах. Осуществление таких процедур может потребовать наличие таких технических средств, которые не рассматриваются в этом Приложении.

Объекты, задействованные в процессе связи, могут четко установить используются ли базовые профили защиты Приложения D, профиль защиты в виде подписи Приложения E или данный смешанный профиль защиты путем определения конфиденциально переданных в сообщениях (**tokenOID** и **algorithmOID**; см. также E.8) идентификаторов объектов.

F.5 Требования H.323

Предполагается, что объекты H.323, реализующие этот смешанный профиль защиты, обеспечивают следующие возможности H.323:

- быстрое соединение;
- туннелирование H.245; и
- модель маршрутизации с помощью GK.

F.6 Аутентификация и контроль целостности

В данном Приложении для предоставления сетевых средств защиты используются следующие термины:

- **Аутентификация и контроль целостности**: Это комбинированное сетевое средство защиты, которое обеспечивает целостность сообщения в сочетании с аутентификацией пользователя. Пользователь аутентифицирует правильность цифровой подписи некоторого количества данных посредством личного ключа или правильность использования соответствующего общего "ключа". Кроме того, сообщение защищается от искажения его данных. Оба сетевых средства защиты обеспечиваются посредством одного и того же механизма защиты. Комбинация аутентификации и контроля целостности возможна только при посегментном сценарии.

ПРИМЕЧАНИЕ. – При применении цифровых подписей может быть обеспечено и такое сетевое средство защиты как защита от неподтверждения. Его применение также зависит от установок битов использования ключей в ключе подписи в сертификате (см. также RFC 3280).

Далее описываются процедуры, используемые в этом профиле.

Процедура IV базируется на цифровых подписях с использованием пары личных/открытых ключей и применением методов симметричной криптографии для обеспечения аутентификации и контроля целостности сообщений RAS, Q.931 и H.245. Терминалы могут использовать этот метод в случае необходимости обеспечения эффективной, широкомасштабной защиты.

В зависимости от стратегии защиты, аутентификация может быть односторонней или взаимной (то есть, аутентификация/контроль целостности применяются также и в обратном направлении, что, таким образом, повышает защиту). Предпочтительный режим защиты должен включать взаимную аутентификацию.

Контроллеры доступа, обнаружив отказ при аутентификации и/или контроле целостности сообщения RAS/сообщения о посылке вызова, полученного от терминала/однорангового контроллера доступа, должны будут ответить соответствующим сообщением отказа, указывающим на безуспешность защиты. Это осуществляется посредством установки причины отказа в **securityDenial** или в другой соответствующий код ошибки защиты в соответствии с разделом B.2.2. В зависимости от способности к распознаванию попытки нарушения защиты и наиболее приемлемого пути реагирования на эту попытку, контроллер доступа, получив защищенное сообщение **xRQ** с неопределенными идентификаторами объектов (**tokenOID**, **algorithmOID**), должен будет ответить незащищенным **xRJ** и отклонить это сообщение с причиной, установленной в **securityDenial**, или же он может сбросить это сообщение. Конечная точка должна будет сбросить полученное незащищенное сообщение, сделать перерыв и снова повторить попытку путем выбора различных OID. Подобным же образом, контроллер доступа, получая защищенное сообщение о посылке вызова SETUP H.225.0 с неопределенными идентификаторами объектов (**tokenOID**, **algorithmOID**), должен будет ответить незащищенным RELEASE COMPLETE и отклонить это сообщение с причиной, установленной в **securityDenial**, или же он может сбросить это сообщение, тогда как контроллер доступа, получая защищенное сообщение FACILITY H.225.0 с неопределенными идентификаторами объектов (**tokenOID**, **algorithmOID**), должен будет ответить незащищенным FACILITY и причиной, установленной в **undefinedReason**, или же он может сбросить это сообщение. Подобным же образом должно быть зарегистрировано возникшее событие защиты. В качестве составной части возвращенного ответа отправитель может предоставить перечень приемлемых сертификатов в отдельных маркерах с тем, чтобы содействовать выбору получателем соответствующего сертификата.

Имеется явная сигнализация H.235 для указания на использование Процедуры IV и соответствующего механизма защиты, основываясь на значении идентификаторов объектов (см. также F.12) и заполнении полей сообщения. В данной Рекомендации идентификаторы объектов обозначаются буквами (к примеру, "A").

Этот профиль не использует поля ICV H.235. Скорее, контрольные признаки криптографической целостности помещаются в поле **signature** маркера **token** в **cryptoSignedToken**, когда дело касается Приложения E, или же контрольные признаки целостности помещаются в хеш-поля **CryptoToken**, когда дело касается Приложения D.

F.7 Процедура IV

Если для посегментной защиты используется Процедура IV, то необходимо придерживаться следующих процедур. Эти процедуры объединяют Процедуру I из Приложения D (см. D.6.3.2) и Процедуру II из Приложения E (см. пункт E.5).

Для первого сообщения, включая соответствующий ответ, посланный в каждом направлении, должна использоваться Процедура II Приложения E (посегментная аутентификация и контроль целостности, см. пункт E.5) со следующими установками:

- OID "A1" взамен OID "A" и OID "S1" взамен OID "S". Использование таких OID позволяет идентифицировать смешанный профиль защиты.
- **algorithmOID** в **tokenOID** должен быть установлен в "W", указывая на использование подписи RSA-SHA1.
- **signature** должно будет содержать подпись RSA, кодированную в ASN.1 (см. пункт E.10).
- **certificate** должно будет содержать сертификат отправителя, если же он отсутствует, то – получателя; **type** должно будет содержать OID "W", указывающий на включенный сертификат RSA-SHA1, или OID "P" (см. пункт E.18), указывающий на то, что сертификат содержит URL.

В сценарии с одним административным доменом "первое сообщение/ответ" определяется как равноценное исходному сообщению RAS/ответу H.225.0; это обычно GRQ/GCF или RRQ/RCF. В сценарии со множеством административных доменов, первое сообщение/ответ внутри каждого домена определяется как указано выше; первое сообщение между доменами определяется как SETUP.

В любом случае передачи в сообщении цифрового сертификата, принимающий объект должен будет сопоставить идентификатор отправителя с идентификатором сертификата в соответствии с процедурой в E.12 с тем, чтобы предотвратить попытки нарушения защиты со стороны посторонних лиц.

Отправитель и получатель производят обмен сообщениями и вычисляют битовую строку аутентифицированного ключа Диффи-Хеллмана. В таблице D.4 дан пример параметров группы Диффи-Хеллмана и рекомендовано употребление с целью защиты, где только это возможно, 1024-битового исходного ключа. Секретный ключ Диффи-Хеллмана должен быть вычислен для каждого участка маршрута, невзирая на то, используется профиль шифрования речевых сообщений или нет.

Исходя из общей битовой строки, вычисляемой обеими сторонами, обе стороны выделяют 160-битовый секретный ключ, используя 160 наименее значащих битов. Результирующий 160-битовый секретный ключ действует как пароль/общий "ключ", используемый в Приложении D.

В сценарии с наличием контроллера доступа в определенных административных доменах отправитель и получатель должны будут использовать два маркера в каждом направлении для сообщений о посылке вызова H.225.0:

- Один **ClearToken** внутри **CryptoToken**, который используется для вычисления мультимедийного ключа, общего для терминалов (см. D.7.1). Он необходим только в случае применения шифрования речевых сообщений.
- Отдельный **ClearToken** используется для вычисления ключа канала связи, который является общим для передающего и принимающего объектов и используется для защиты звена сигнализации. Этот ключ канала связи замещает общий для контроллеров доступа в Приложении D пароль. **TokenOID** этого **ClearToken** должен быть установлен в "Q", указывая на использование схемы Диффи-Хеллмана и смешанного профиля защиты. Вычисление ключа канала связи происходит таким же образом, как и вычисление мультимедийного ключа (см. D.7.1).

ПРИМЕЧАНИЕ 1. – В условиях прямой маршрутизации передающий/принимающий объекты и терминалы корреспондируются. В условиях маршрутизации с помощью GK ключ канала связи совместно, посегментно используется каждой парой одноранговых контроллеров доступа, тогда как мультимедийный ключ совместно используется из конца в конец.

В условиях маршрутизации с помощью GK этот GK должен будет направлять полученный маркер Диффи-Хеллмана от конечной точки к следующему сегменту.

Для всех самых первых сообщений/ответов, посылаемых в каждом направлении, должна использоваться Процедура I из Приложения D (см. D.6.3.2). Это приемлемо также в таких сценариях, когда множество контроллеров доступа размещено в рамках административного домена. В этом случае нет необходимости в управлении асимметричными ключами; достаточно следовать Приложению D.

Это приложение может быть использовано с системами H.235, версии 1, если дело касается ограниченного использования sendersID и generalID, как описано в пункте E.17.

Ожидается, что контроллер доступа получит из конкретной фиксированной конечной точки только одно сообщение **RRQ**, включая маркер DH с цифровой подписью. Однако, потерянные или задержанные сообщения **RCF/RRJ** могут привести к повторению передачи с использованием другого подписанного **RRQ**.

В случае, если соответствующий ответ регистрации не поступает вовремя в конечную точку, то конечная точка может повторить попытку. Для этого, эта конечная точка должна будет использовать самый последний маркер DH, а не новый порядковый номер и новую временную метку.

Для конкретной фиксированной конечной точки контроллер доступа должен будет использовать самое последнее полученное подписанное сообщение **RRQ** и вывести из этого маркера DH общий "ключ", невзирая на то, имеет ли уже этот GK в наличии этот общий "ключ". Таким образом, GK должен будет перезаписать любой имеющийся общий "ключ", заменив его заново вычисленным ключом. GK должен будет ответить подписанным сообщением **RCF**, содержащим маркер DH ответа. Предпочтительно, чтобы маркер DH ответа был создан заново.

ПРИМЕЧАНИЕ 2. – Рекомендуемый и предпочитаемый метод обновления ключей состоит в использовании сообщения FACILITY как описано в пункте F.9. Однако, установлено, что обновление ключей может осуществляться путем использования другого дополнительного подписанного сообщения **RRQ** с новым маркером DH.

ПРИМЕЧАНИЕ 3. – Контроллер доступа, владеющий общим "ключом" должен будет ответить на HMAC-защищенное сообщение **RRQ** (согласно Приложению D) HMAC-защищенным сообщением ответа.

F.8 Ассоциация защиты при одновременных вызовах

Необходимо обеспечивать оптимизацию в том случае, когда фиксированная пара объектов должна будет обрабатывать несколько независимых одновременных вызовов, используя один канал передачи сигналов вызова. Вместо создания нескольких ключей канала связи с использованием ключей Диффи-Хеллмана для каждого, определяется ассоциация защиты, которая охватывает множество одновременных вызовов.

Более точно, ассоциация защиты охватывает все вызовы между фиксированной парой объектов во время действия канала передачи сигналов вызова. Объекты используют флаг **multipleCalls** в рамках SETUP для указания на возможность передачи сигналов множества вызовов через одно соединение (см. 7.3/H.323).

Если используется одно соединение сигнализации, то необходимо сформировать только один общий ключ канала связи (см. рисунок F.1).

С другой стороны, если флаг **multipleCalls** в рамках SETUP не установлен, то ключ канала связи должен быть заново отдельно вычислен для каждого вызова.

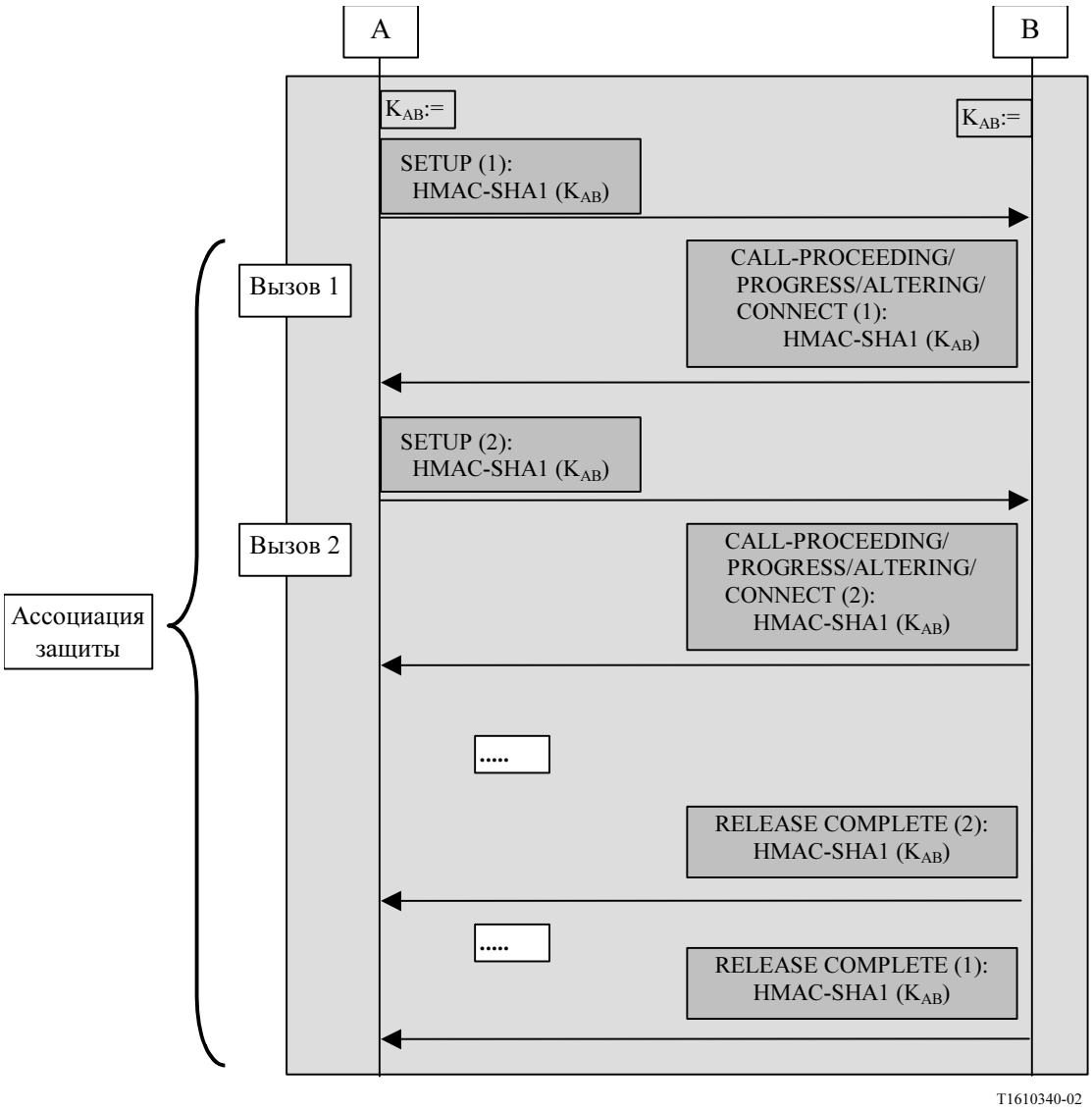


Рисунок F.1/H.235 – Ассоциация защиты при одновременных вызовах

F.9 Обновление ключей

Дополнительная процедура обновления ключей позволяет любому объекту, участвующему в процессе связи (GK или терминал), заменить используемый сеансовый ключ на новый. Такое обновление ключа должно инициироваться любым объектом, который чувствует в этом потребность. Обновление ключа может быть мотивировано несанкционированно раскрытым сеансовым ключом, пониманием того, что сеансовый ключ перестал или перестанет быть секретным, или другими критериями стратегии защиты. Все эти аспекты выходят за рамки данной Рекомендации.

Инициатор вызывает обновление ключа, используя сообщение FACILITY. Сообщение FACILITY для обновления ключа передает новый маркер Диффи-Хеллмана, дополнительный цифровой сертификат и цифровую подпись инициатора. По получении сообщения FACILITY, получатель отвечает подобным же сообщением FACILITY, передавая маркер Диффи-Хеллмана, дополнительный цифровой сертификат и цифровую подпись получателя. По завершении процедуры обновления ключей, инициатор и получатель должны использовать вычисленный новый ключ канала связи.

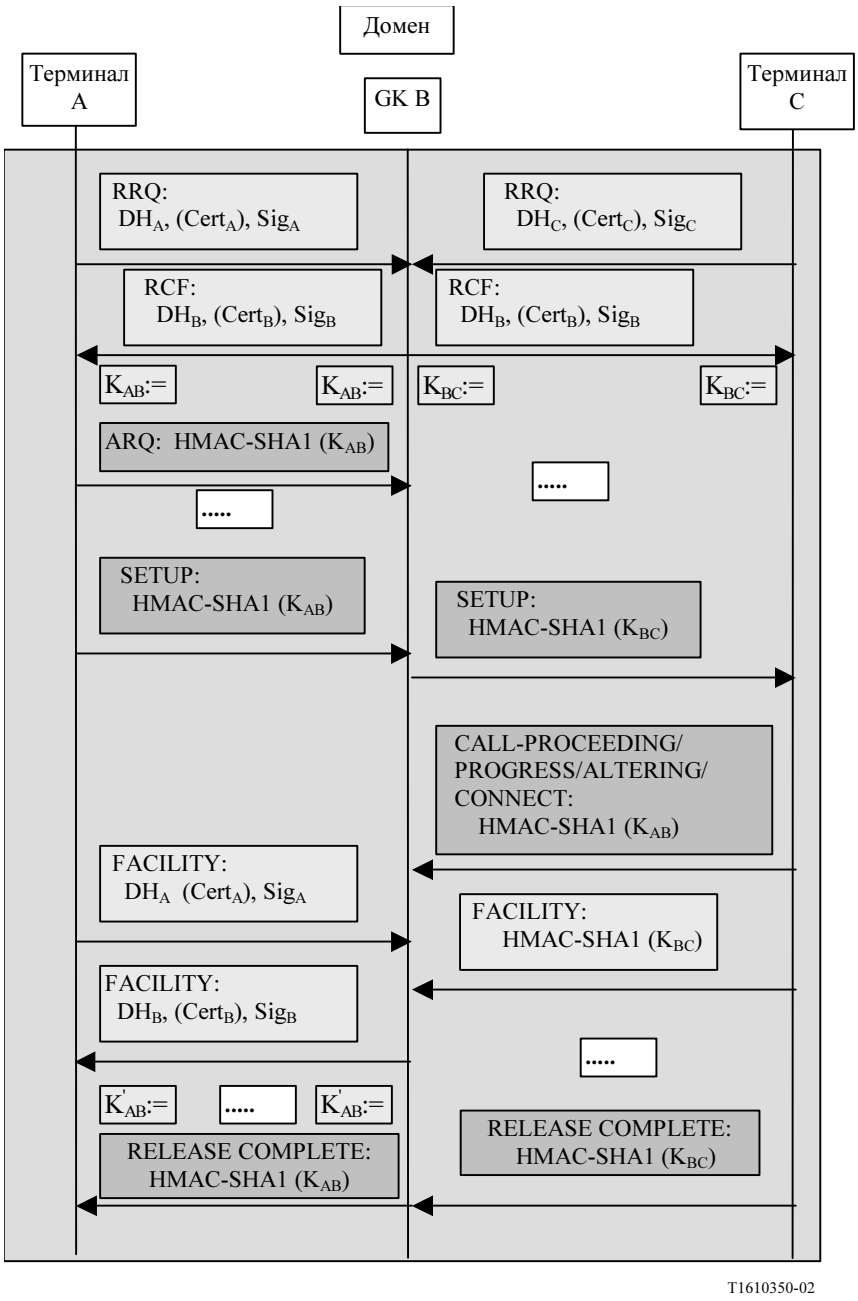
- **tokenOID** маркера **ClearеToken** в рамках FACILITY должен быть установлен в "Q", указывая на использование маркера Диффи-Хеллмана и смешанного профиля защиты. Вычисление ключа канала связи происходит тем же способом, что и вычисление мультимедийного сеансового ключа (см. D.7.1).

Сообщение FACILITY, в целях обновления ключа, должно быть защищено согласно Процедуре II Приложения E. Любое другое сообщение FACILITY, не переносящее маркер Диффи-Хеллмана, не должно применяться с целью обновления ключей и должно быть защищено согласно Процедуре I Приложения D.

F.10 Иллюстративные примеры

На структурных диаграммах на рисунках F.2 и F.3 показано использование Приложения F применительно к основному потоку сообщений. Заметим, что на этих диаграммах не показан весь поток сообщений и что несколько сообщений опущены с целью упрощения. Сообщения, отмеченные светло-серым цветом, относятся к профилю подписи в Приложении E, тогда как сообщения, отмеченные темно-серым цветом, относятся к базовому профилю в Приложении D. Цифры указывают на (наиболее важные) элементы защиты каждого сообщения (H.235 CryptoTokens, Tokens), опуская детали.

На структурной диаграмме на рисунке F.2 показан основной поток сообщений в сценарии с одним контроллером доступа в рамках отдельного административного домена. Допуская, что сертификат контроллера доступа известен всем задействованным сертификатам и что терминалам также известен сертификат контроллера доступа, нет необходимости в передаче сертификатов во время процедуры регистрации.



Cert	Сертификат пользователя	K, K'	симметричный ключ канала связи
DH_A	Маркер Диффи-Хеллмана g^a mod p	Sig	цифровая подпись
DH_B	Маркер Диффи-Хеллмана g^b mod p		
EP	Конечная точка (терминал)		
GK	Контроллер доступа		

Рисунок F.2/H.235 – Структурная диаграмма при отдельном административном домене

ПРИМЕЧАНИЕ 1. – Рисунки F.2 и F.3 также охватывают процедуру быстрого старта, если сообщения о посылке вызова SETUP и CALL PROCEEDING/PROGRESS/ALERTING/CONNECT включают маркер быстрого старта (см. 8.1.7/H.323). В ином случае, допускается режим небыстрого старта, согласно 7.3.1/H.323. На рисунке F.2 также показана процедура обновления ключей между терминалом A и контроллером доступа B, используя FACILITY.

На рисунке F.3 показан примерный поток сообщений в сценарии с различными административными доменами. При том, что смешанный профиль защиты применяется внутри каждого домена между терминалом и контроллером доступа, что показано на рисунке F.2, смешанный профиль защиты может применяться также между обоими доменами на этапе установления соединения.

ПРИМЕЧАНИЕ 2. – На рисунке F.3 опущен процесс связи между граничными элементами (BE) и процесс связи между GK-BE. На рисунке F.3 также показана процедура обновления ключей между обоими доменами с использованием FACILITY.

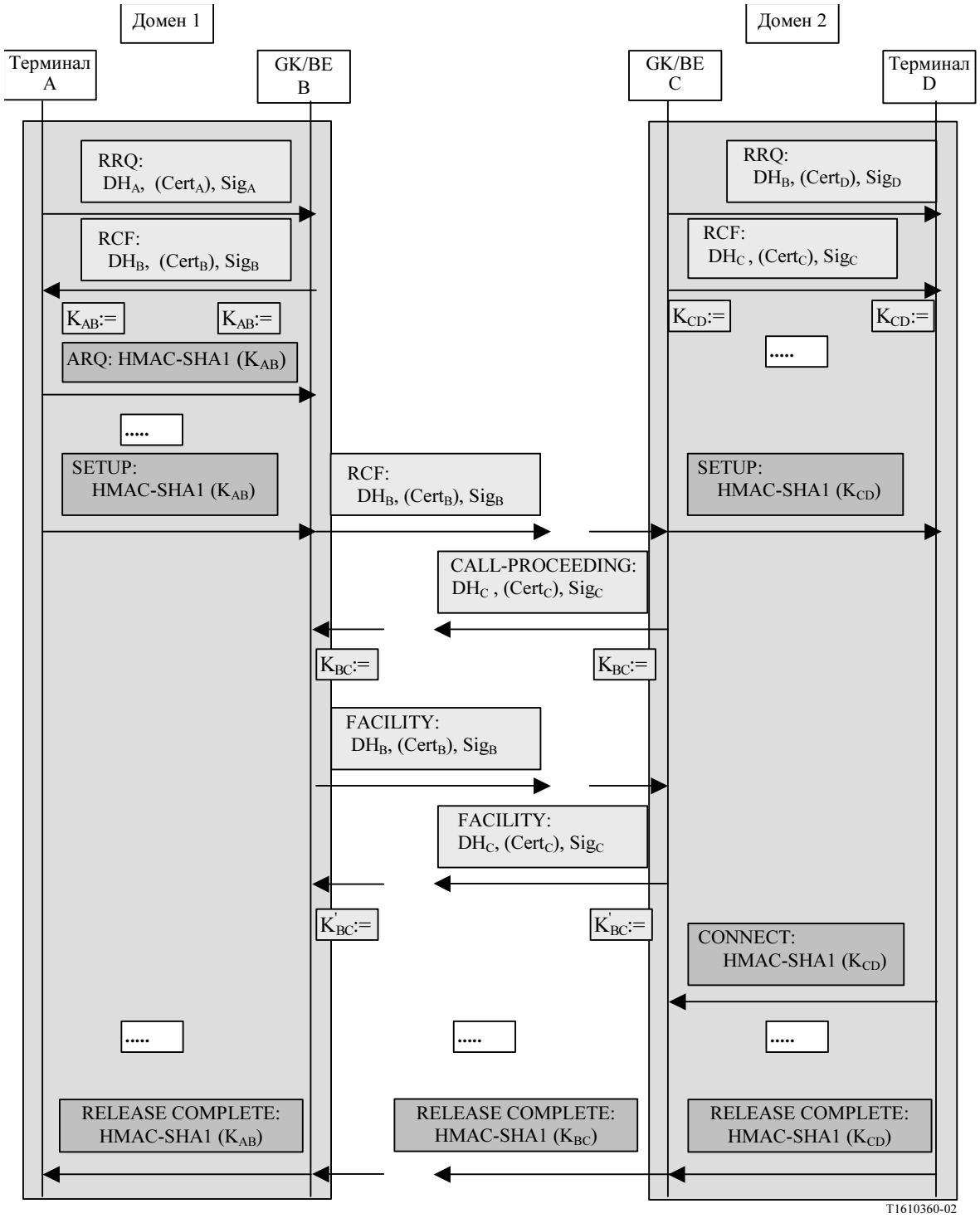


Рисунок F.3/H.235 – Структурная диаграмма при множестве административных доменов

F.11 Режим многоадресной передачи

Многоадресные сообщения H.225.0, такие как **GRQ** или **LRQ**, должны включать **CryptoToken** согласно Процедуре II, причем **generalID** не установлен. Когда такие сообщения посылаются одноадресно, тогда это сообщение должно включать **CryptoToken** с установленным **generalID**.

F.12 Перечень защищенных сигнальных сообщений

Процедура IV использует Процедуру I в Приложении D или Процедуру II в Приложении E в зависимости от сценария и фактического сообщения, как показано внизу.

F.12.1 RAS H.225.0

Сообщение RAS H.225.0	Поля сигнализации H.235	Аутентификация и контроль целостности	Защита от неподтверждения
GatekeeperRequest, GatekeeperConfirm, GatekeeperReject, если задействуется выявленный GK RegistrationRequest, RegistrationConfirm, RegistrationReject, если задействуется выявленный GK	CryptoToken, ClearToken	Процедура II	Процедура II
Любое другое сообщение RAS (Прим. 2)	CryptoToken	Процедура I	
ПРИМЕЧАНИЕ 1. – Для одноадресных сообщений должна применяться Процедура II с полями защиты в используемом **CryptoToken**. ПРИМЕЧАНИЕ 2. – Сообщение о выявлении GK и многоадресные сообщения не посылаются.			

F.12.2 Сообщение о посылке вызова H.225.0 (отдельный административный домен)

Сообщение о посылке вызова H.225.0	Поля сигнализации H.235	Аутентификация и контроль целостности	Защита от неподтверждения
Setup-UUIE, Connect-UUIE (Примечание 1) Facility-UUIE (Примечание 2), Alerting-UUIE, CallProceeding-UUIE, Facility-UUIE, Progress-UUIE, Information-UUIE, ReleaseComplete-UUIE, Status-UUIE, StatusInquiry-UUIE, SetupAcknowledge-UUIE, Notify-UUIE	CryptoToken, ClearToken	Процедура I	
Facility-UUIE (Примечание 3)	CryptoToken	Процедура II	Процедура II
ПРИМЕЧАНИЕ 1. – Допуская, что любое сообщение является первым в каждом направлении. ПРИМЕЧАНИЕ 2. – Не используется для обновления ключей. ПРИМЕЧАНИЕ 3. – Используется для обновления ключей.			

F.12.3 Сообщение о посылке вызова H.225.0 (множество административных доменов)

Сообщение о посылке вызова H.225.0	Поля сигнализации H.235	Аутентификация и контроль целостности	Защита от неподтверждения
Setup-UUIE, Connect-UUIE (Примечание 1) Alerting-UUIE (Примечание 2), CallProceeding-UUIE, Facility-UUIE, (Примечание 3) Progress-UUIE, Information-UUIE, ReleaseComplete-UUIE	CryptoToken, ClearToken	Процедура II	Процедура II
Alerting-UUIE (Примечание 4), CallProceeding-UUIE, Facility-UUIE (Примечание 5), Progress-UUIE, Information-UUIE, ReleaseComplete-UUIE, Status-UUIE, StatusInquiry-UUIE, SetupAcknowledge-UUIE, Notify-UUIE	CryptoToken	Процедура I	Процедура I
ПРИМЕЧАНИЕ 1. – Допуская, что любое сообщение является первым в каждом направлении. ПРИМЕЧАНИЕ 2. – Любое из этих сообщений оказывается первым в любом направлении. ПРИМЕЧАНИЕ 3. – Используется для обновления ключей. ПРИМЕЧАНИЕ 4. – Ни одно из этих сообщений не оказывается первым в любом направлении. ПРИМЕЧАНИЕ 5. – Не используется для обновления ключей.			

F.13 Перечень идентификаторов объектов

В таблице F.2 перечислены в идентификаторы объектов OID, которые упоминаются в тексте.

Таблица F.2/H.235 – Идентификаторы объектов, используемые в Приложении F

Опорное значение идентификатора объекта	Значение(я) идентификаторов объектов	Описание
"A 1"	{itu-t (0) recommendation (0) h (8) 235 version (0) 2 20}	Используется для замещения OID "A" в Процедуре II Приложения E для CryptoToken-tokenOID, указывая на то, что хеш-величина/подпись RSA включают все поля в RAS H.225.0 или в сообщении о посылке вызова (аутентификация и контроль целостности).
"S 1"	{itu-t (0) recommendation (0) h (8) 235 version (0) 2 21}	Используется для замещения OID "S" в Процедуре II Приложения E для ClearToken-tokenOID, указывая на то, что ClearToken используется для аутентификации сообщений и контроля целостности. Этот OID в сквозном CryptoToken явно указывает на применение также маркера DH во время быстрого старта.
"Q"	{itu-t (0) recommendation (0) h (8) 235 version (0) 2 22}	Используется в Процедуре IV, указывая на то, что ClearToken при последовательном соединении переносит маркер DH.
"W"	{itu-t (0) recommendation (0) h (8) 235 version (0) 2 23}	Используется в Процедуре IV в качестве OID алгоритма, указывая на использование цифровой подписи, основывающейся на RSA-SHA1.

Приложение G

Использование протокола защищенной передачи данных в режиме реального времени (SRTP) в сочетании с протоколом управления ключами MIKEY в Рекомендации H.235

Это Приложение подлежит дальнейшему изучению.

Приложение H

Управление ключами RAS

Это Приложение подлежит дальнейшему изучению.

Приложение I

Обеспечение вызовов с прямой маршрутизацией

I.1 Область рассмотрения

Цель данного Приложения – предоставить рекомендации по процедурам защиты для применения передачи сигналов о посылке вызова с прямой маршрутизацией в сочетании с профилями защиты D и F H.235.

Данный профиль защиты предлагается в качестве дополнительного и может служить дополнением к профилям защиты Приложений D или F.

В этом Приложении представлены подробности реализации для пункта B.6, используя методы управления симметричными ключами.

ПРИМЕЧАНИЕ. – В этом Приложении показаны процедуры защиты при упрощенном сценарии, что дает возможность дальнейших разработок более совершенных общих процедур защиты; этот вопрос подлежит дальнейшему изучению.

I.2 Введение

При применении модели маршрутизации с помощью контроллера доступа часто обращаются к H.323. К примеру, применение этой модели обеспечивает наиболее оптимальный биллинг и другие функциональные возможности. Все более широкое применение моделей маршрутизации с помощью контроллера доступа служит также причиной того, что в рамках Рек. МСЭ-Т H.235 различные профили защиты (такие как в Приложениях D, E, F) описываются с ориентацией на эту модель передачи вызовов.

Однако, при том, что существует необходимость обеспечения растущего количества параллельных каналов, применение модели прямой маршрутизации с помощью контроллера доступа сможет привести к лучшим показателям эксплуатационного качества и возможностям масштабирования. Преимущество этой модели состоит в использовании контроллера доступа для регистрации, доступа, адресной резолюции и управления полосой частот при осуществлении установления соединения непосредственно между конечными точками в сквозном режиме передачи.

В этом Приложении описываются усовершенствования базового профиля защиты из Приложения D и смешанного профиля защиты Приложения F, произведенные для обеспечения прямой маршрутизации вызовов с помощью контроллера доступа.

I.3 Условные обозначения

Идентификаторы объектов в данном тексте обозначаются символически (к примеру, "I1"), в пункте I.12 представлен перечень фактических цифровых значений для символических идентификаторов объектов, см. также пункт 5.

I.4 Термины и определения

Для данной Рекомендации определения, данные в пункте 3 Рек. МСЭ-Т H.323, H.225.0, H.235, и X.800, применяются соотносительно с определениями в данном пункте.

I.5 Символы и аббревиатуры

В этом Приложении используются следующие аббревиатуры:

$\{M\}_{K;\,S,\,IV}$ Шифрование *M* в режиме EOFB с использованием секретного ключа *K*, секретного ключа с "привязками" *S* и вектора инициализации *IV*

CT ClearToken

DRC Вызов с прямой маршрутизацией

EPID Идентификатор конечной точки

GKID Идентификатор контроллера доступа

K_{AG} Общий "ключ" (Приложение D, Приложение F) между EP A и GK G

K_{BG} Общий "ключ" (Приложение D, Приложение F) между EP B и GK G

KS_{AG} Секретный общий ключ "с привязками" между EP A и GK G

KS_{BG} Секретный общий ключ "с привязками" между EP B и GK G

K'_{AG} Общий ключ шифрования между EP A и GK G

K'_{BG} Общий ключ шифрования между EP B и GK G

K_{AB} Общий ключ шифрования между EP A и EP B

I.6 Нормативные источники ссылок

Следующие Рекомендации МСЭ-Т и другие источники содержат положения, которые, будучи упомянутыми в качестве ссылок в данном тексте, составляют положения данной Рекомендации. На момент публикации указанные издания были действующими. Все Рекомендации и другие источники ссылок подлежат пересмотру; поэтому всем пользователям этих Рекомендаций предлагается рассмотреть возможность использования самого последнего издания этих Рекомендаций и других источников ссылок, перечисленных ниже. Перечень действующих рекомендаций МСЭ-Т публикуется регулярно. Ссылка на любой документ в рамках данной Рекомендации не придает ему, даже при том, что это отдельный документ, статуса Рекомендации.

– Рекомендация МСЭ-Т H.225.0 (2003 г.), *Протоколы передачи сигналов вызовов и пакетирование потоков мультимедийной информации для мультимедийных систем связи в пакетном режиме.*

– Рекомендация МСЭ-Т H.323 (2003 г.), *Мультимедийные системы связи в пакетном режиме.*

– Рекомендация МСЭ-Т X.800 (1991 г.), *Архитектура защиты для Взаимосвязи Открытых Систем для приложений МККТТ.*

– ИСО/МЭК 10118-3: 2004, *Информационные технологии – Методы защиты – Хеш-функции – Часть 3: Специальные хеш-функции.*

– Стандарт IETF RFC 2104 (1997 г.), *HMAC: Хеширование ключей для аутентификации сообщений.*

– Стандарт IETF RFC 2246 (1999 г.), *Протокол TLS, версия 1.0.*

I.7 Общее описание

Базовый профиль защиты Приложения D (см. основной текст этой Рекомендации), а также смешанный профиль защиты Приложения F (см. Приложение F) (после первого квитирования) применяют общий "ключ" для обеспечения аутентификации и/или контроля целостности при посегментном сценарии, используя в качестве доверительного промежуточного узла контроллер доступа. Используя модель прямой маршрутизации вызовов, невозможно обеспечить общий "ключ" между двумя конечными точками. Также непрактично применять заранее устанавливаемый общий "ключ" для защиты связи, так

как в этом случае все конечные точки будут заранее осведомлены о том, какая другая конечная точка будет вызвана.

В этом приложении рассматривается сценарий, показанный на рисунке I.1, при котором конечные точки закреплены за единым контроллером доступа и применяют передачу сигналов вызова с прямой маршрутизацией. Этот сценарий предполагает наличие незащищенной IP сети в зоне этого контроллера.

Предполагается, что каждая конечная точка имеет взаимосвязь и ассоциацию защиты с контроллером доступа и что каждая конечная точка надежно зарегистрирована этим контроллером доступа, используя или базовый профиль защиты, или смешанный профиль защиты.

Таким образом, контроллер доступа способен обеспечивать общий "ключ" для прямо сообщающихся конечных точек, используя метод, подобный методу Кербера (Kerberos).

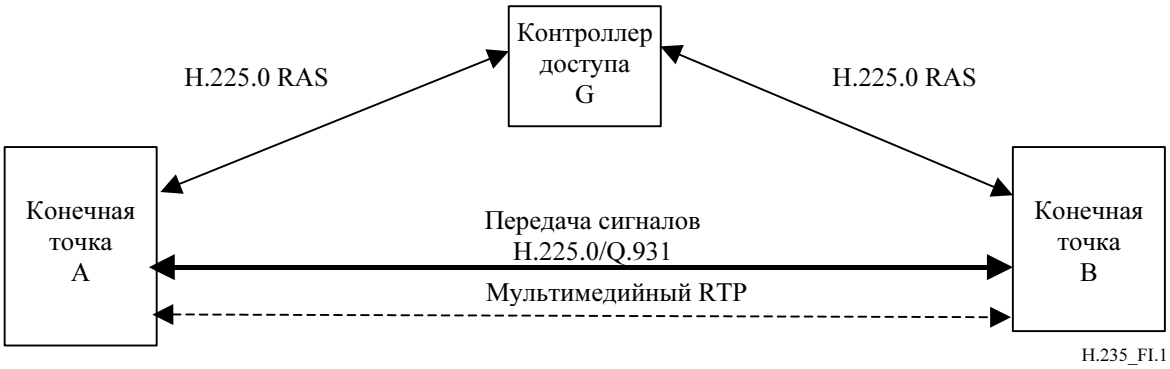


Рисунок I.1/H.235 – Сценарий с прямой маршрутизацией вызова

I.8 Ограничения

В данном Приложении не рассматриваются сценарии с прямой маршрутизацией, при которых конечные точки закреплены за определенными контроллерами доступа. Кроме того, в этом Приложении не рассматриваются сценарии с прямой маршрутизацией, но без контроллера доступа. Все эти вопросы подлежат дальнейшему изучению.

I.9 Процедура DRC

Конечные точки, способные обеспечить этот профиль защиты, должны указывать на это во время передачи GRQ и/или RRQ путем включения отдельного ClearToken с **tokenOID**, установленным в "I0"; все остальные поля в этом ClearToken не должны использоваться. Контроллер доступа, наделенный возможностями, указанными в Приложении I и желающий предоставить эти функциональные возможности, должен будет ответить соответствующим GCF. RCF с отдельным ClearToken и включенным в него **tokenOID**, установленным в "I0", и все другие поля в ClearToken остаются неиспользованными.

До того, как конечная точка A начнет посылать сообщения о посылке вызова непосредственно к другой конечной точке B, эта конечная точка A или B должна обратиться за доступом к контроллеру доступа G, используя ARQ. Конечная точка A должна будет включать в **ARQ** отдельный ClearToken с **tokenOID**, установленным в "I0", а все стальные поля в ClearToken остаются неиспользованными.

Контроллер доступа, определив, что эти конечные точки A и B соответствуют данному приложению, должен сформировать данные ключа и ClearTokens, как указано ниже.

Контроллер доступа способен вычислить основывающийся на вызове общий "ключ" K_{AB} помимо обычной команды ARQ. Затем, этот основывающийся на вызове общий "ключ" передается к обеим конечным точкам, используя ClearTokens. Эти ClearTokens передаются в рамках сообщения ACF и отсылаются обратно к вызывающему объекту.

Должны быть включены два ClearTokens – один CT_A для вызывающего объекта A, а другой CT_B для вызываемого объекта B. Каждый **ClearToken** должен содержать OID ("I1" или "I2") в рамках **tokenOID**, который указывает на то, предназначен ли этот маркер вызывающему объекту (OID "I1" для CT_A) или вызываемому объекту (OID "I2" для CT_B).

ClearToken, как определено в этом Приложении, может использоваться в сочетании с другими профилями защиты, такими как в Приложении D или в Приложении F, в которых также используются **ClearTokens**. В таком случае ClearToken Приложения I должен также использовать и поля тех других **ClearToken**. К примеру, для того, чтобы использовать Приложение I в сочетании с Приложением D, должны присутствовать поля **timestamp**, **random**, **generalID**, **sendersID**, и **dhkey** и они должны использоваться так, как описано профилями защиты Приложения D.

ID контроллера доступа (GKID) должен размещаться в рамках **sendersID**, тогда как **generalID** должен содержать идентификатор конечной точки A (CT_A) или конечной точки B (CT_B).

K' обозначает ключ шифрования, который является общим для конечной точки и GK. Ключи шифрования K'_{AG} и K'_{BG} для шифруемого сквозного ключа K_{AB} должны быть получены из общего "ключа" контроллера доступа и конечных точек (K_{AG} или K_{BG}), используя основывающуюся на PRF процедуру деривации ключа, как описано в пункте I.10, при этом **keyDerivationOID** в **V3KeySyncMaterial** должен содержать "Annex I-HMAC-SHA1-PRF", см. пункт I.12.

Контроллер доступа должен формировать общий сеансовый ключ K_{AB}, который совместно используется конечной точкой A и конечной точкой B.

Сеансовый ключ K_{AB} должен быть зашифрован посредством K'_{AG} (для CT, предназначенного для конечной точки A) или же посредством K'_{BG} (для CT, предназначенного для конечной точки B), используя любой алгоритм шифрования.

Усовершенствованный режим шифрования OFB (EOFB) (см. B.2.5) должен использоваться с секретным, ориентированным на конечную точку, ключом с "привязками". К приемлемым алгоритмам шифрования относятся (см. D.11):

- DES (56 битов) в режиме EOFB, использует OID "Y1": является необязательным;
- 3DES (168 битов) в открытом режиме EOFB, использует OID "Z1": является необязательным;
- AES (128 битов) в режиме EOFB, использует OID "Z2": используется по умолчанию и является рекомендуемым;
- RC2-совместимый (56 битов) в режиме EOFB, использует OID "X1": является необязательным.

При режиме шифрования EOFB GK должен генерировать случайное исходное значение IV. При OID "X1", OID "Y1" и OID "Z1" IV имеет 64 бита и должен передаваться в рамках **iv8** в **params** внутри **V3KeySyncMaterial**; когда же IV имеет 128 битов при OID "Z2", то он должен передаваться в рамках **iv16** в **params** внутри **V3KeySyncMaterial**.

Полученный зашифрованный текст $\{K_{AB}\}_{K'AG,\ KSAG,\ IV}$ соответственно $\{K_{AB}\}_{K'BG,\ KSBG,\ IV}$ должен затем передаваться в структуре данных **h235key** как часть **secureShareSecret,** при этом он должен быть размещен в рамках **encryptedSessionKey** структуры данных **secureSharedSecret**. Алгоритм шифрования должен быть указан в **algorithmOID** ("X1", "Y1", "Z1" или "Z2") в рамках **V3KeySyncMaterial**.

Для ClearToken, предназначенного для конечной точки A, идентификатор конечной точки B ($EPID_B$) должен быть помещен внутри **generalID** в **V3KeySyncMaterial**. Подобным же образом, для ClearToken, предназначенного для конечной точки B, идентификатор конечной точки A ($EPID_A$) должен быть помещен внутри **generalID** в **V3KeySyncMaterial**.

Для алгоритмов шифрования EOFB не должен использоваться **encryptedSaltingKey**.

Контроллер доступа должен включать в сообщение **ACF** в направлении конечной точки A оба ClearToken – CT_A и CT_B.

Конечная точка A должна идентифицировать CT_A путем проверки **tokenOID** "I1" в рамках ClearToken.

Конечная точка A должна подтвердить новизну CT_A путем проверки **timestamp**. Для дополнительного контроля защиты необходимо проверить **generalID** и **sendersID** в ClearToken и **generalID** в рамках **V3KeySyncMaterial**. Если полученный CT_A был проверен и его новизна подтверждена, то конечная точка A должна осуществить поиск IV и вычислить K'_{AG} и KS_{AG} способом, описанным выше для контроллера доступа. Конечная точка A должна будет расшифровать данные **encryptedSessionKey**, выявленные в рамках **V3KeySyncMaterial** в CT_A с тем, чтобы получить K'_{AB}.

Если была подтверждена новизна полученного $CT_{A,}$ то конечная точка A может послать сообщение SETUP к конечной точке B. Это сообщение SETUP включает CT_B. Сообщение SETUP должно быть защищено (аутентифицировано и/или его целостность защищена) согласно Приложению D или

Приложению F, используя K_{AB} в качестве приемлемого общего "ключа". Для этого, **generalID** в хешированном ClearToken (но не CT_B!) из Приложения D должен быть установлен в $EPID_B$.

Конечная точка B должна идентифицировать CT_B путем проверки **tokenOID** "I2" в рамках ClearToken.

Конечная точка B должна подтвердить новизну полученного CT_B путем проверки **timestamp**. Для дополнительного контроля защиты необходимо проверить **generalID** и **sendersID** в ClearToken и **generalID** в рамках **V3KeySyncMaterial**. Если была подтверждена новизна полученного CT_B, то конечная точка B должна осуществить поиск IV и вычислить K'_{BG} и KS_{BG} способом, описанным выше для контроллера доступа. Конечная точка B должна расшифровать данные **encryptedSessionKey**, выявленные в рамках **V3KeySyncMaterial** в CT_B с тем, чтобы получить K'_{AB}.

В случае, если была подтверждена новизна CT_B, конечная точка B в состоянии продолжить передачу сигналов вызова путем приемлемого ответа CALL-PROCEEDING, ALTERING или CONNECT и т. п. В случае, если установлено, что CT_B не новый, или же проверка защиты сообщения SETUP оказалась безуспешной, конечная точка B должна ответить RELEASE-COMPLETE, а **ReleaseCompleteReason** должна быть установлена в значение ошибка защиты, как указано в B.2.2.

Когда применяется мультимедийная защита (см. пункт D.7), то конечная точка A и конечная точка B должны будут обменяться полуключами Диффи-Хеллмана согласно D.7.1 и сформировать динамический, ориентированный на сеанс основной ключ, исходя из которого можно затем получить ориентированные на среду передачи сеансовые ключи.

На рисунке I.2 показан основной коммуникационный поток:

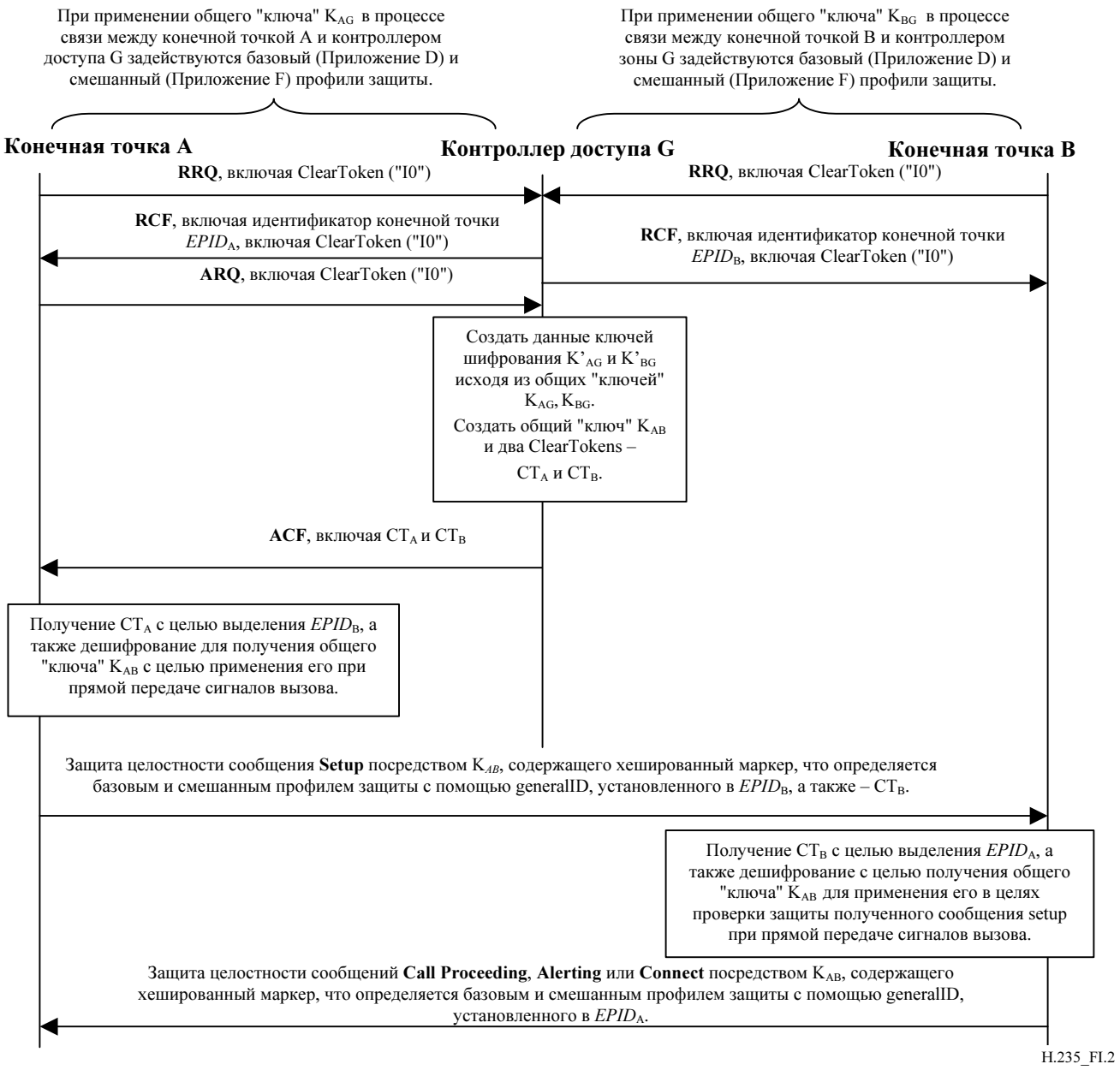


Рисунок I.2/H.235 – Основной коммуникационный поток

I.10 Процедура деривации ключей на основе PRF

В этом пункте описывается процедура, определяющая способ деривации данных ключа исходя из общего "ключа" и других параметров.

Ключ шифрования K'_{AG} должен вычисляться с использованием PRF (см. пункт B.7) при том, что параметр *inkey* устанавливается в K_{AG}, а *label* должна быть установлена в постоянное значение 0x2AD01C64 || **challenge**.

Подобным же образом, ключ шифрования K'_{BG} должен вычисляться с использованием PRF, при этом параметр *inkey* устанавливается в K_{BG}, а *label* должна быть установлена в постоянное значение 0x1B5C7973 || **challenge**. В обоих случаях *outkey_len* должна быть установлена равной значению требуемой длины ключа шифрования для выбранного алгоритма шифрования.

Используя эту же PRF, необходимо, чтобы контроллер доступа или каждая конечная точка формировали общий секретный ключ с "привязками". Этот ключ с "привязками", при использовании его в режиме шифрования EOFB, защищает от попыток воздействия на нешифрованный текст CT_B со стороны EP A, в ином случае, EP A может попытаться выявить K_{BG}.

KS_{AG} обозначает общий секретный ключ с "привязками", который совместно используется EP A и GK G. KS_{AG} должен вычисляться, используя PRF с параметром *inkey*, установленным в $K_{AG,}$ а *label* должна устанавливаться в постоянное значение 0x150533E1 || **challenge**. KS_{BG} должен вычисляться с использованием PRF с параметром *inkey*, установленным в K_{BG}, а *label* должна устанавливаться в постоянное значение 0x39A2C14B || **challenge**.

ПРИМЕЧАНИЕ. – 32-битовые постоянные целые числа (т. е. 0x2AD01C64 и т. п.) получаются из десятичных чисел *e* (т. е. 2.7182...), если каждая постоянное целое число состоит из девяти десятичных чисел (к примеру, первые девять десятичных чисел 718281828 = 0x2AD01C64). Последовательности первых девяти десятичных чисел выбираются не произвольно, а в виде последовательных "порций" из десятичных чисел *e*.

I.11 Процедура деривации ключа на основе FIPS-140

В этом пункте могла бы рассматриваться процедура, описывающая способ деривации данных ключа из общего "ключа" и других параметров, используя соответствующий FIPS-140 модуль криптографии. Но этот вопрос подлежит дальнейшему изучению.

I.12 Перечень идентификаторов объектов

Таблица I.1/H.235 – Идентификаторы объектов, используемые в Приложении I H.235

Опорное значение идентификатора объекта	Значение идентификатора объекта	Описание
"I0"	{itu-t (0) recommendation (0) h (8) 235 version (0) 3 48}	Используется в процедуре DRC во время GRQ/RRQ, GCF/RCF и ARQ, чтобы дать возможность EP/GK указать на соответствие Приложению I.
"I1"	{itu-t (0) recommendation (0) h (8) 235 version (0) 3 49}	Используется в процедуре DRC для ClearToken tokenOID, указывая на то, что ClearToken содержит сквозной ключ для вызывающего объекта.
"I2"	{itu-t (0) recommendation (0) h (8) 235 version (0) 3 50}	Используется в процедуре DRC для ClearToken tokenOID, указывая на то, что ClearToken содержит сквозной ключ для вызываемого объекта.
"Annex I-HMAC-SHA1-PRF"	{itu-t (0) recommendation (0) h (8) 235 version (0) 3 51}	Используется в процедуре DRC для keyDerivationOID в рамках V3KeySyncMaterial для индикации используемого метода деривации ключа в I.10, применяя псевдослучайную функцию HMAC-SHA1.

Дополнение I

Подробное описание реализации H.323

I.1 Методы заполнения зашифрованного текста

На страницах 191 и 196 [Schneier] представлено описание процесса принудительного "захвата" зашифрованного текста. На рисунках I.1–I.5 показаны сами методы.

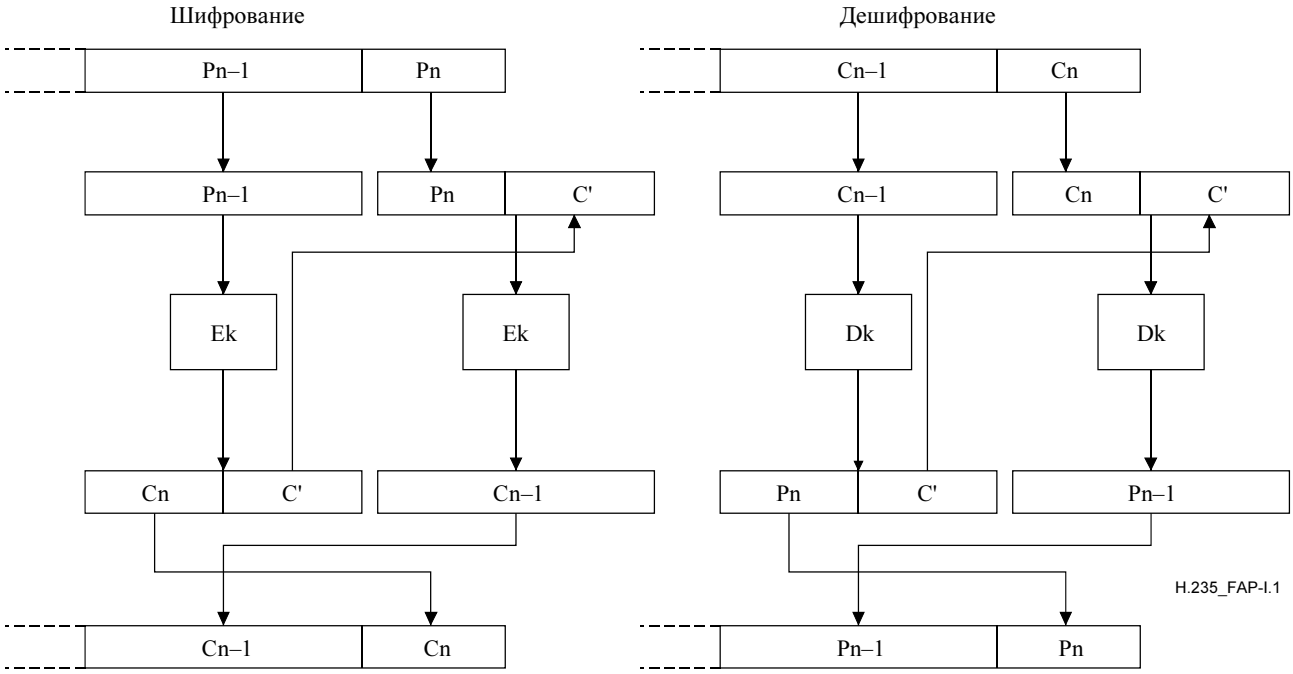


Рисунок I.1/H.235 – Принудительный "захват" зашифрованного текста в режиме ECB

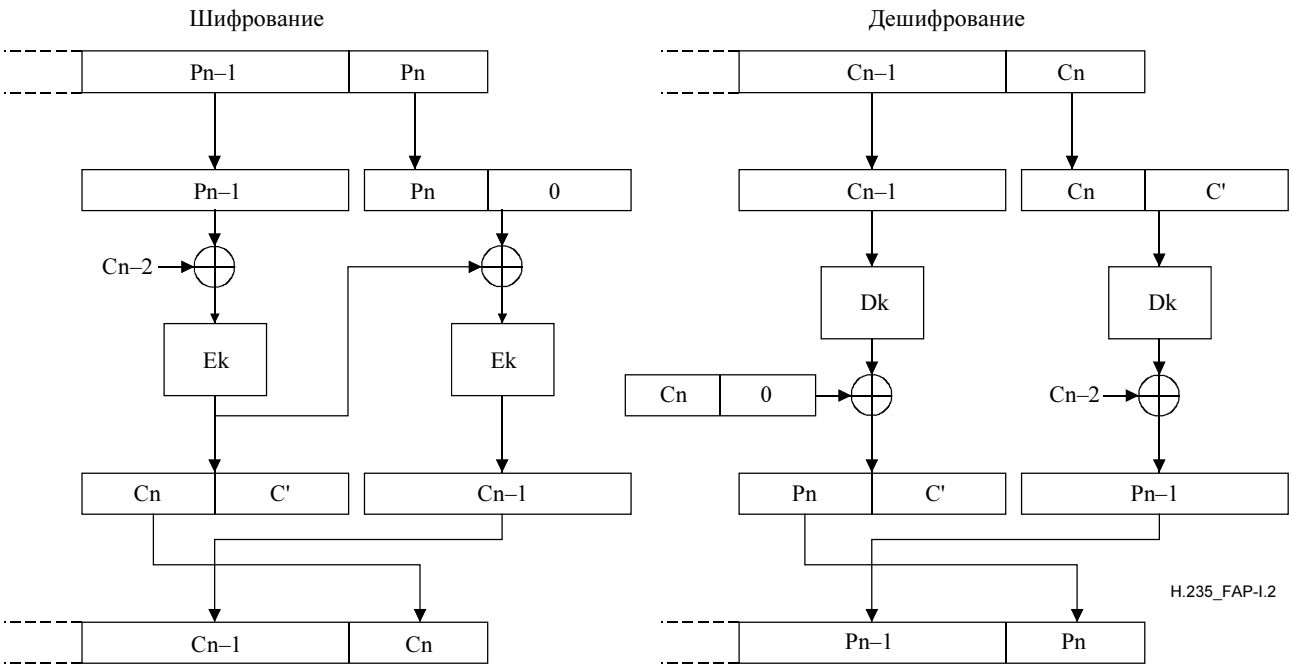


Рисунок I.2/H.235 – Принудительный "захват" зашифрованного текста в режиме CBC

ПРИМЕЧАНИЕ. – Принудительный "захват" зашифрованного текста в режимах ECB или CBC требует наличия в полезной нагрузке, по крайней мере, одного полного блока. Реализации, применяющие принудительный "захват" зашифрованного текста в режиме ECB или CBC, должны удостовериться, что в полезной нагрузке всегда переносится, по крайней мере, один криптографический блок; к примеру, посредством правильного выбора частоты выборки/пакетирования или алгоритма шифрования.

В случае, если полезная нагрузка охватывает менее одного отдельного блока, исходный вектор (IV) должен использоваться в качестве предыдущего блока шифрованного текста, если в режиме CBC присутствует режим принудительного "захвата" зашифрованного текста.

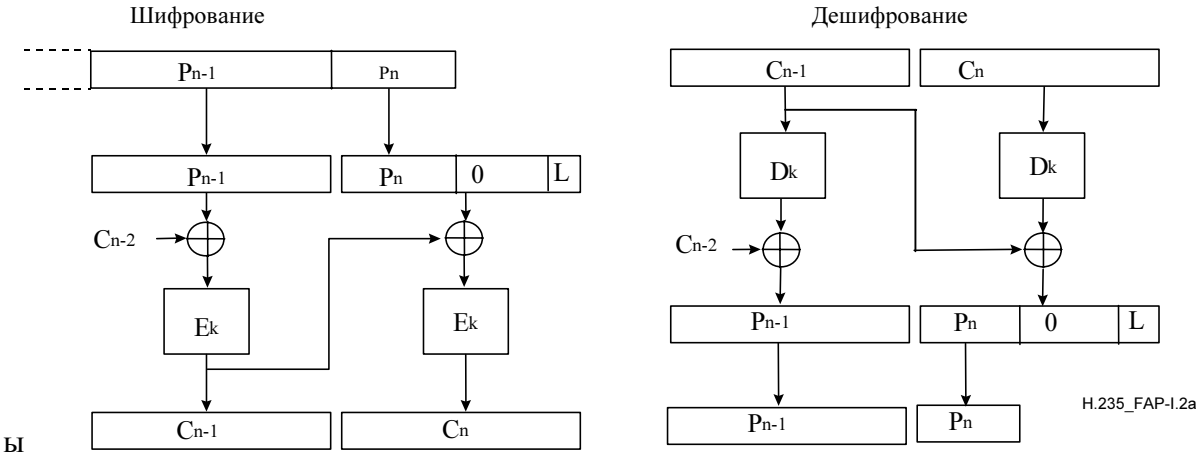


Рисунок I.2a/H.235 – Заполнение нулями в режиме CBC

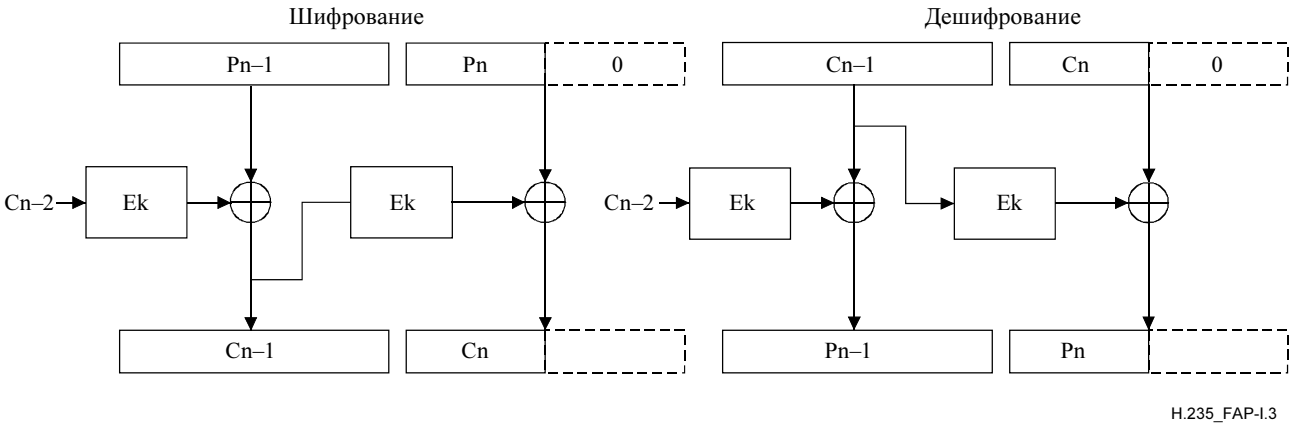
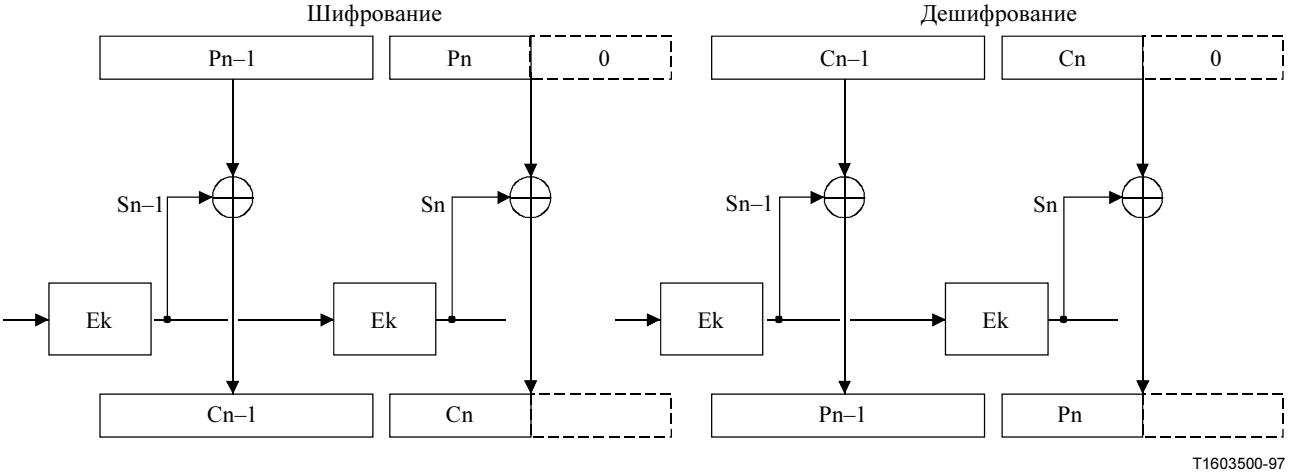


Рисунок I.3/H.235 – Заполнение нулями в режиме CFB



ПРИМЕЧАНИЕ. – S_i – это результат повторяющегося шифрования (т. е. пермутации) IV.

Рисунок I.4/H.235 – Заполнение нулями в режиме OFB

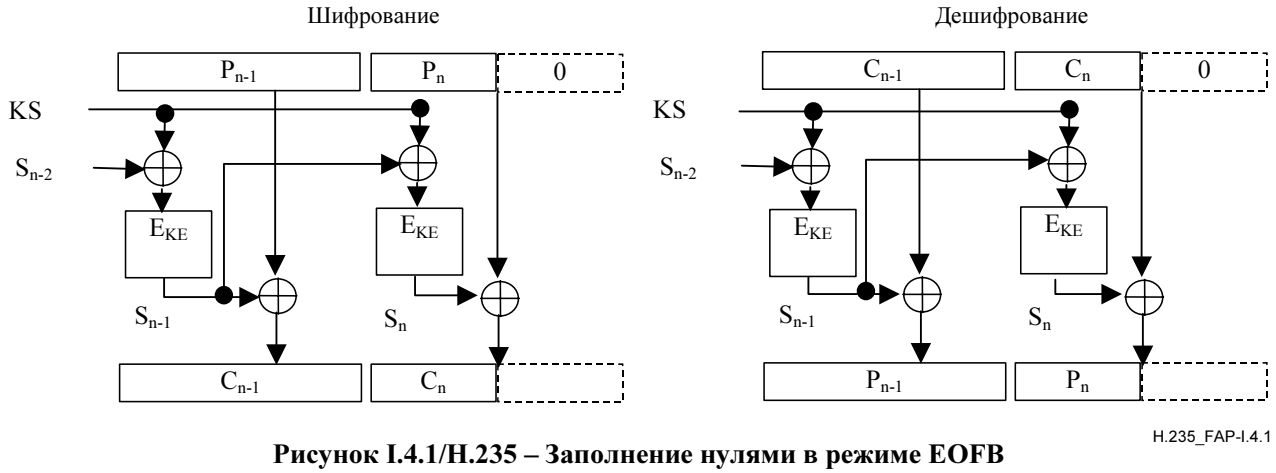


Рисунок I.4.1/H.235 – Заполнение нулями в режиме EOFB

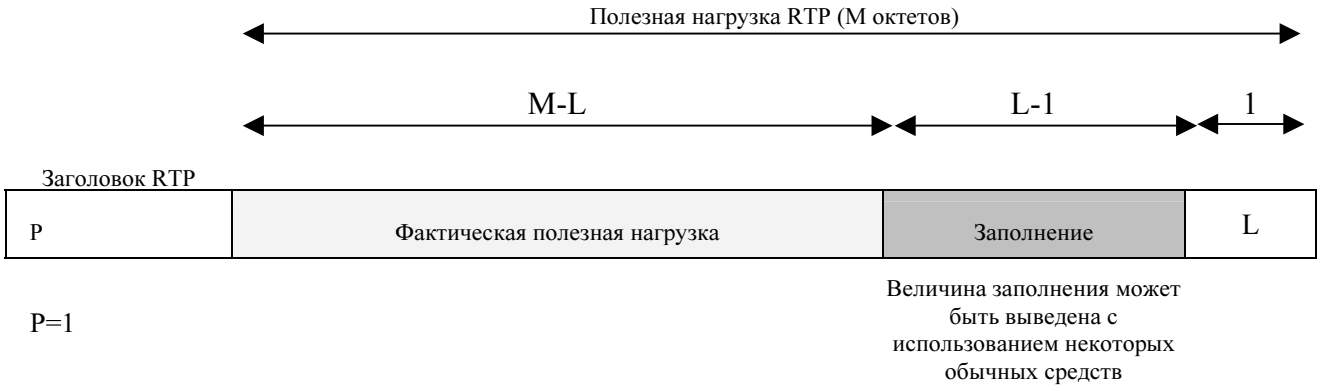


Рисунок I.5/H.235 – Заполнение, предписываемое RTP H.235_FRA-1.5

I.2 Новые ключи

Процедуры, описанные в 8.5/H.323, выполняются MC, чтобы исключить какого-либо участника из конференции. Ведущий терминал может создавать новые ключи шифрования для логических каналов (и не передавать их исключенной стороне); это можно использовать для того, чтобы воспрепятствовать мониторингу потоков мультимедийной информации со стороны исключенного участника.

I.3 Элементы H.323

Обычно MC(U), шлюзы и контроллеры доступа (при реализации модели маршрутизации с помощью контроллера доступа) являются доверительными по отношению к секретности канала управления. Если канал установления соединения (H.225.0) защищен *и* маршрутизирован через контроллера доступа, то он также должен считаться доверительным. Если любой из этих компонентов H.323 должен работать с потоками мультимедийной информации (т. е. производить смешивание, транскодирование) тогда, по определению, они также должны быть доверительными для секретности мультимедийной информации.

Защитные прокси (хотя они и не являются элементами H.323) могут также считаться доверительными, так как они завершают соединения, и, весьма возможно, что им придется обрабатывать сообщения и потоки мультимедийной информации.

I.4 Примеры реализаций

В следующих подпунктах описываются примеры реализаций, которые могли бы быть разработаны в рамках H.235. Они не предназначены для ограничения многочисленных возможностей, предоставляемых настоящей Рекомендации, а скорее просто служат более конкретными примерами применения Рек. МСЭ-Т H.323.

I.4.1 Маркеры

В настоящем пункте описан пример использования маркеров защиты для того, чтобы спрятать или скрыть адресную информацию назначения. Приведен сценарий с конечной точкой, которая желает вызвать другую конечную точку, используя ее хорошо известный псевдоним. Точнее, в данном процессе участвуют конечная точка H.323, контроллер доступа, шлюз традиционной телефонной сети (POTS-gateway) и телефон, что показано на рисунке I.6.

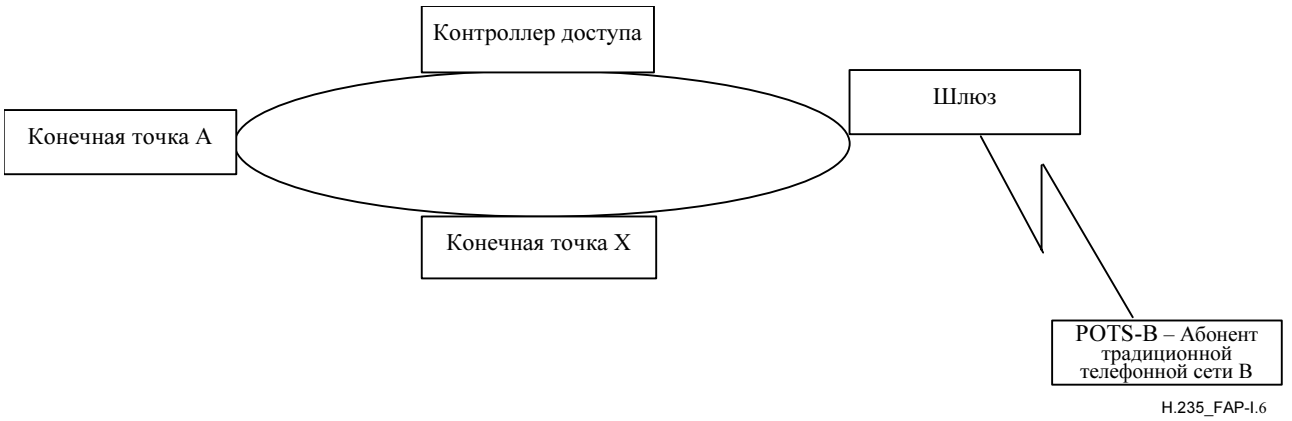


Рисунок I.6/H.235 – Маркеры

На данном рисунке сеть H.323 может функционировать аналогично телефонной сети с идентификатором (ID) вызывающего абонента. Этот сценарий иллюстрирует ситуацию, при которой *вызывающий абонент* не хочет раскрывать свой физический адрес, но в то же время разрешает завершить вызов. Это может быть важно в шлюзах H.323 традиционной телефонной сети, в которой конкретный номер телефона, возможно, должен оставаться скрытым.

Предположим, что EPA пробует вызывать абонента традиционной телефонной сети (POTS-B), и POTS-B не хочет раскрывать свой номер телефона (E.164) EPA. (Вопрос разработки подобной стратегии выходит за рамки настоящего примера.)

- EPA должна послать ARQ своему контроллеру доступа с тем, чтобы получить адрес телефона POTS в том виде, как он представлен посредством своего псевдонима/шлюза. Контроллер доступа должен опознать его как "личный" псевдоним, учитывая, что для завершения соединения он должен возвратить адрес шлюза POTS (подобно возврату адреса шлюза H.320, если конечная точка H.320 вызывается конечной точкой H.323).
- В возвращенном сообщении ACF контроллер доступа возвращает, как и ожидалось, адрес шлюза POTS. Адресная информация, которая требуется для того, чтобы позвонить на оконечный телефон (то есть, номер телефона), возвращается в зашифрованном маркере, включенном в ACF. Этот зашифрованный маркер содержит фактический E.164 (номер) телефона, который не может быть ни расшифрован, ни понят вызывающим абонентом (т. е. EPA).

- Конечная точка посылает сообщение SETUP к шлюзу (чей адрес передачи сигналов вызова был возвращен в ACF), включив в него непрозрачный маркер(ы), который был получен в ACF.
- Шлюз, по получении SETUP, посылает свой ARQ своему контроллеру доступа, включая в него любой маркер(ы), которые были получены в SETUP.
- Контроллер доступа способен расшифровать маркер(ы) и вернуть номер телефона в ACF.

Часть ASN.1 из примерной структуры маркера показана ниже с описанием содержимого полей. Предположим, что мы используем **cryptoEncodedGeneralToken**, чтобы записать зашифрованный номер телефона.

Любая реализация могла бы выбрать **tokenOID**, обозначающий этот маркер как содержащий номер телефона E.164. Конкретный метод, используемый для зашифровки этого номера телефона (например, 56-битовый DES), должен быть включен в определение "ENCRYPT" поля **algorithmOID**.

```
CryptoToken::= CHOICE
{
     cryptoEncodedGeneralToken SEQUENCE  -- Маркер, ориентированный на
                                         -- универсальное применение
     {
          tokenOID OBJECT IDENTIFIER,
          ENCRYPTED { EncodedGeneralToken }
     },
.
.
. [сокращенный текст]
.

}
```

CryptoToken должен будет передаваться в сообщении SETUP (от EPA к GW) и в сообщениях **ARQ** (от GW к контроллеру доступа), как было показано выше. После того, как контроллер доступа расшифровал маркер (номер телефона), он должен передать его нешифрованную версию в **clearToken**.

I.4.2 Использование маркеров в системах H.323

Существуют некоторые недоразумения в части использования отдельных **CryptoH323Tokens** при передаче их в сообщениях RAS. Имеются две главных категории **CryptoH323Tokens**: используемые для процедур H.235 и используемые зависимым от приложения способом. Применение этих маркеров должно соответствовать следующим правилам:

- Все определенные H.235 поля (например, **cryptoEPPwdHash**, **cryptoGKPwdHash**, **cryptoEPPwdEncr**, **cryptoGKPwdEncr**, **cryptoGKCert** и **cryptoFastStart**) должны будут использоваться с процедурами и алгоритмами, как описано в настоящей Рекомендации.
- При зависимом от приложения или частном использовании маркеров для обмена ними должен использоваться **nestedCryptoToken**.
- Любой используемый **nestedCryptoToken** должен иметь **tokenOID** (идентификатор объекта), однозначно его идентифицирующий.

I.4.3 Использование случайных значений H.235 в системах H.323

Случайное значение, которое передается в последовательности сообщений xRQ/xCF между конечными точками и контроллерами доступа, может быть обновлено контроллером доступа. Как описано в B.4.2, это случайное значение может обновляться в любом сообщении xCF, для использования его в последующих сообщениях xRQ, передаваемых от конечной точки. Вследствие того, что сообщения RAS могут быть потеряны (включая сообщения xCF/xRQ), обновленное случайное значение также может быть потеряно. Исправить эту ситуацию можно путем повторной инициализации контекста защиты, но это должно определяться конкретной реализацией.

Реализации, которые требуют наличия множества ожидающих обработки запросов RAS, будут ограничиваться обновлением случайных значений, используемых при любом виде аутентификации. Если обновление значения происходит при каждом ответе на запрос, то параллельные запросы невозможны. Единственно возможным решением является наличие логического "окна", во время которого случайное значение остается постоянным. Эта проблема определяется конкретной реализацией.

I.4.4 Пароль

В этом примере предполагается, что пользователь является абонентом контроллера доступа (то есть пользователь должен находиться в его зоне) и имеет соответствующий идентификатор подписки и пароль. Пользователь должен быть зарегистрирован контроллером доступа, используя идентификатор подписки (который передается в псевдониме – H323ID) и зашифровывая строку вызова, предоставляемую контроллером доступа. При этом подразумевается, что контроллер доступа также знает пароль, соответствующий идентификатору подписки. Контроллер доступа должен аутентифицировать пользователя, проверяя правильность шифрования строки вызова.

Ниже приведен пример процедуры регистрации с аутентификацией контроллера доступа:

1) Если для обнаружения контроллера доступа конечная точка использует **GRQ**, то одним из псевдонимов в этом сообщении станет идентификатор подписки (как **H323ID**). Поле а**uthenticationcapability** должно будет содержать **AuthenticationMechanism** поля **pwdSymEnc**, **algorithmOID** должны быть установлены таким образом, что бы обозначать полный набор алгоритмов шифрования, которые поддерживает данная конечная точка. (Например, одним из них может быть 56-битовый DES в режиме EBC.)

2) Контроллер доступа должен будет ответить сообщением **GCF** (при условии, что он распознал псевдоним), переносящим элемент **tokens**, который содержит один **ClearToken**. Данный **ClearToken** должен будет включать элементы **challenge** и **timeStamp**. Элемент **challenge** должен будет содержать 16 октетов. (Для предотвращения повторных попыток нарушения защиты **ClearToken** должен содержать **timeStamp**.) Поле **authenticationmode** должно быть установлено в **pwdSymEnc**, а **algorithmOID** должен быть установлен таким образом, чтобы указывать на алгоритм шифрования, требуемый контроллером доступа (например, 56-битовый DES в режиме EBC).

   Если контроллер доступа не поддерживает ни один из **algorithmOID**, указанных в **GRQ**, то тогда он должен будет ответить сообщением **GRJ,** содержащем **GatekeeperRejectReason**, в **resourceUnavailable**.

3) Приложение конечной точки должно затем попытаться зарегистрироваться в GK (в одном из них), который ответил сообщением **GCF**, посылая ему сообщение **RRQ,** содержащее **cryptoEPPwdEncr** в **cryptoToken**s. Поле **cryptoEPPwdEncr** должно иметь **algorithmOID** алгоритма шифрования, который был согласован при обмене **GRQ**/**GCF**, и зашифрованный вызов.

   Ключ шифрования создается из пароля пользователя пароля посредством процедуры, описанной в 10.3.2. Затем результирующая "строка" октетов используется как ключ алгоритма DES, для шифрования **challenge**.

4) Когда контроллер доступа получает зашифрованный вызов в сообщении **RRQ**, то он должен сравнить его с идентично сформированным зашифрованным вызовом, чтобы аутентифицировать регистрирующегося пользователя. Если две зашифрованные строки не соответствуют друг другу, то контроллер доступа должен ответить сообщением **RRJ** с полем **RegistrationRejectReason**, установленным в **securityDenial**, или другим соответствующим кодом ошибки, согласно B.2.2. Если они соответствуют друг другу, то контроллер доступа посылает конечной точке сообщение **RCF**.

5) Если контроллер доступа получает сообщение **RRQ**s которое не содержит приемлемый элемент **cryptoTokens**, то он должен ответить сообщением **RRJ** с **GatekeeperRejectReason** в **discoveryRequired**. Конечная точка, по получении такого **RRJ** может произвести обнаружение, которое позволит контроллеру доступа или конечной точке обменяться новым вызовом.

   ПРИМЕЧАНИЕ. – Сообщение **GRQ** может быть однонаправленным – к контроллеру доступа.

I.4.5 Защита на уровне Интернет-протокола (IPSEC)

В целом IPSEC [IPSEC] может использоваться для обеспечения аутентификации и, дополнительно, конфиденциальности (то есть шифрования) на IP-уровне, прозрачном для любого (прикладного) протокола, который реализуется на более высоком уровне. Для этого прикладной протокол не должен актуализироваться; актуализации подлежит только стратегия защиты на каждом конце.

Например, чтобы максимально эффективно использовать IPSEC при простом соединении "точка–точка", необходимо использовать следующий сценарий:

1) Вызывающая конечная точка и ее контроллер доступа должны определять стратегию, требующую использования IPSEC (аутентификация и, дополнительно, конфиденциальность) на основе протокола RAS. Таким образом, прежде чем первое сообщение RAS посылается от конечной точки к контроллеру доступа, программа "демон" ISAKMP/Oakley в конечной точке должна согласовать сетевые средства защиты, которые должны быть использованы для пакетов, передаваемых в известный порт канала RAS и из него. Как только согласование закончено, канал RAS будет работать точно таким же образом, как если бы он не был защищен. При использовании этого защищенного канала контроллер доступа должен будет сообщить конечной точке адрес и номер порта канала передачи сигналов вызова в вызываемой конечной точке.

2) По получении адреса и номера порта этого канала передачи сигналов вызова, вызывающая конечная точка должна будет динамически обновить свою стратегию защиты, чтобы потребовать необходимую защиту IPSEC для этого адреса и пару "протокол–порт". Теперь, когда вызывающая конечная точка делает попытки связаться с этим адресом/портом, пакеты должны будут ставиться в очередь на то время, пока между конечными точками проводится согласование по ISAKMP/Oakley. По завершении этого согласования, для адреса/порта будет установлена Ассоциация защиты IPSEC (Security Association), и может продолжиться передача сигналов Q.931.

3) При обмене сообщениями Q.931 SETUP и CONNECT конечные точки могут согласовывать использование IPSEC для канала H.245. Это позволит конечным точкам снова динамически актуализировать свои базы данных по вопросам стратегии IPSEC, чтобы форсировать использование IPSEC на этом соединении.

4) Как и в случае с каналом передачи сигналов вызова, прозрачное согласование ISAKMP/Oakley должно проводиться до передачи каких-либо пакетов H.245. Аутентификация, осуществленная при этом обмене ISAKMP/Oakley, станет начальной попыткой аутентификации пользователь–пользователь и послужит установлению (вероятно) защищенного канала между этими двумя пользователями, чтобы согласовывать по нему характеристики аудиоканала. Если после некоторого определенного запроса/аутентификации, какой-то из пользователей не удовлетворен этой аутентификацией, то могут быть выбраны иные сертификаты, и обмен ISAKMP/Oakley может быть повторен.

5) После каждой аутентификации ISAKMP/Oakley H.245 происходит обмен новыми данными ключа для аудио-канала RTP. Эти данные ключа распределяются ведущим терминалом по защищенному каналу H.245. Поскольку для ведущего терминала определен протокол H.245 для распределения данных мультимедийного ключа по каналу H.245 (чтобы дать возможность многоточечной связи), не рекомендуется использование IPSEC для канала RTP.

Зашифрованный канал H.245 является потенциальной проблемой для прокси или брандмауэра с трансляцией сетевых адресов (NAT firewall), так как в протоколе H.245 передаются динамически присвоенные номера портов. Такие брандмауэры должны, чтобы правильно функционировать, расшифровывать, изменять и заново шифровать протокол. По этой причине, в Рек. МСЭ-Т H.245 был введен "защищенный" логический канал. Если используется этот канал, то канал H.245 может оставаться незащищенным; аутентификация и создание ключей будут осуществляться посредством "защищенного" логического канала. При передаче сигналов по логическому каналу этот канал должен быть защищен с помощью IPSEC, а общий "ключ", используемый этим "защищенным" логическим каналом, должен будет использоваться для защиты элемента **EncryptionSync**, распространяемого ведущим терминалом по каналу H.245.

I.4.6 Поддержка работы внутренних серверов

Внутренние серверы (BES) являются важной дополнительной функцией для всей основывающейся на H.323 мультимедийной среды. Например, BES обеспечивают услуги аутентификации пользователей, авторизации служб, а также – учета, тарификации, биллинга и другие услуги. В простой модели такие услуги может обеспечивать контроллер доступа. При сложной архитектуре GK не всегда может обеспечивать такие услуги, потому что он не имеет доступа к базам данных BES или же, потому что он может быть частью другого административного домена. Кроме того, терминал или пользователь обычно не знают свои BES.

На рисунке I.7 показан сценарий с мультимедийным терминалом (например, SASET), контроллером доступа и связанными с ними BES. Точное описание процесса взаимодействия BES с контроллером доступа выходит за рамки рассмотрения Рек. МСЭ-Т H.323. Тут применимы несколько методов и

протоколов: RADIUS (Система дистанционной аутентификации пользователей по телефонным линиям) (см. RFC 2865), которая рассматривается как одна из наиболее важных систем и которая широко используется поставщиками услуг.

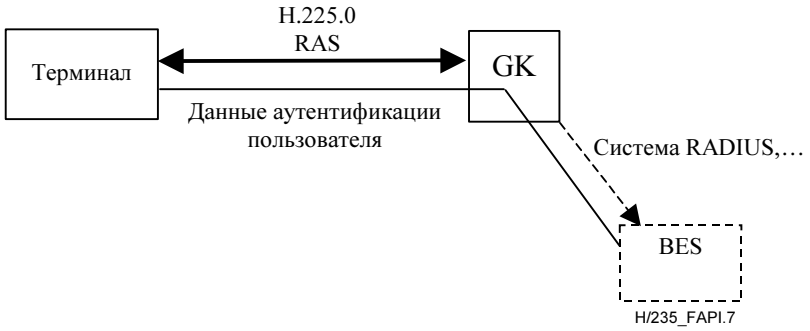


Рисунок I.7/H.235 – Сценарий с внутренним сервером

GK, предлагающий поддержку BES, должен, по крайней мере, поддерживать два следующих режима:

1) **Режим работы по умолчанию**, при котором терминал не знает BES и требует доверительных отношений с GK. Этот терминал посылает данные аутентификации пользователя в зашифрованной форме (**cryptoEncryptedToken**) к GK, который дешифрует их, извлекает информацию об аутентификации пользователя и отправляет ее к BES. Основывающиеся на пароле шифрование **ClearToken** выполняется с применением особого "ключа", который известен терминалу и GK, в **CryptoToken**. Ключ шифрования может быть извлечен из пароля, с помощью которого терминал защищенно регистрируется в GK.

   CryptoToken передает **cryptoEncryptedToken**, в котором **tokenOID** установлен в "M", указывая на режим работы BES; **token** содержит:

   - **algorithmOID**, указывающий на алгоритм шифрования; "Y" (DES56-CBC), "Z" (3DES-ocbc); см. D.11;
   - **paramS** – не используется;
   - **encryptedData**, установленный равным октетному представлению зашифрованного **ClearToken**.

   ClearToken содержит в виде **password** данные аутентификации пользователя. К защищенным данным **ClearToken** могут относиться – пароль/PIN, идентификатор пользователя, номер карточки предварительной оплаты услуг и номер кредитной карточки. Поле **timeStamp** устанавливается в значение, равное текущему значению времени терминала, **random** содержит равномерно возрастающий порядковый номер, **sendersID** устанавливается в значение, равное идентификатору терминала, а **generalID** – в значение идентификатора GK. Исходное значение алгоритма шифрования должно будет сохраняться постоянным; оно может быть составляющей ключа подписки терминала.

   ПРИМЕЧАНИЕ. – **ClearToken** не передается.

2) **Режим RADIUS**: при этом режиме BES и пользователь терминала совместно используют общий "ключ", и GK не должен быть доверительным для аутентификации BES в режиме RADIUS. GK просто пересылает RADIUS-вызов, полученный от BES в рамках *Access-Challenge* к терминалу и посылает ответ пользователя в качестве RADIUS-ответа в рамках *Access-Request* в обратном направлении. Терминал и GK согласовывают эту характеристику вызова/ответа **radius** в **AuthenticationBES** в рамках **AuthenticationMechanism** во время обнаружения контроллера доступа.

   По получении RADIUS-сообщения *Access-Challenge*, передающего вызов, GK помещает 16-октетный вызов в поле **challenge** в **ClearToken** во время запроса терминала с помощью **GCF** или любого другого сообщения RAS. Значение **tokenOID**, равное 'K', в **ClearToken** указывает на RADIUS-вызов.

   Терминал может затем представить вызов пользователю и ожидать введенного ответа. Этот терминал должен будет ответить сообщением RAS, в котором ответ помещается в поле **challenge**

в **ClearToken**. Значение **tokenOID**, равное 'L', в **ClearToken**, указывает на RADIUS-ответ.

В таблице I.1 перечислены все опорные значения идентификаторов объектов.

Таблица I.1/H.235 – Идентификаторы объектов, используемые в I.4.6

Опорное значение идентифи-катора объекта	Значение идентификатора объекта	Описание
"K"	{itu-t (0) recommendation (0) h (8) 235 version (0) 2 31}	обозначает RADIUS-вызов в ClearToken
"L"	{itu-t (0) recommendation (0) h (8) 235 version (0) 2 32}	обозначает RADIUS-ответ (передается в поле challenge) в ClearToken
"M"	{itu-t (0) recommendation (0) h (8) 235 version (0) 2 33}	обозначает режим работы BES по умолчанию с помощью защищенного пароля в ClearToken

Дополнение II

Подробное описание реализации H.324

Подлежит дальнейшему изучению.

Дополнение III

Подробное описание реализации других терминалов серии H

Подлежит дальнейшему изучению.

Дополнение IV

Библиография

[Daemon] DAEMO N (J.): *Создание функций шифрования и хеширования,* Ph.D. Thesis, Katholieke Universiteit Leuven, March 1995.

[IPSEC] MAUGHAN (D.), SCHERTLER (M.), SCHNEIDER (M.), TURNER (J.): *Интернет-протокол управления ключами и ассоциация защиты сети Интернет (ISAKMP)*, draft-ietf-ipsec-isakmp-08.text, *Специальная группа инженерной поддержки сети Интернет*, 1997 г.

[ИСО/МЭК14888-3] *Информационные технологии – Методы защиты – Цифровые подписи с дополнением; Часть 3: Механизмы на основе сертификатов*, 1998 г.

[J.170] Рекомендация МСЭ-Т J.170 (2002 г.), *Спецификация защиты IPCablecom.*

[MIKEY] ARKKO (J.), CARRARA (E.), LINDHOLM (F.), NASLUND (M.), NORRMAN (K.); *"MIKEY: Манипуляции мультимедийными ключами в сети Интернет"*, Internet Draft <draft-ietf-msec-mikey-06.txt>, RFC xxxx, Work in Progress (MSEC WG), IETF, 02/2003.

{Примечание издателя: Этот стандарт (RFC#) будет включен, когда он появится}

[PKCS] PKCS #1 v2.0: *Криптографический стандарт RSA*; RSA Laboratories; October 1, 1998; http://www.rsa.com/rsalabs/pubs/PKCS/index.html.

[PKCS] PKCS #7: *Синтаксический стандарт криптографических сообщений*, An RSA Laboratories Technical Note, Version 1.5, Revised November 1, 1993; http://www.rsa.com/rsalabs/pubs/PKCS/index.html

[RTP] SCHULZRINNE (H.), CASNER (S.), FREDERICK (R.), JACOBSON (V.): *RTP: Транспортный протокол для приложений реального времени, RFC 3550, Специальная группа инженерной поддержки сети Интернет*, 2003 г.

[Schneier] SCHNEIER (B.): *Прикладная криптография: Протоколы, Алгоритмы, и исходные коды в C*, 2-е издание, John Wiley & Sons, Inc., 1995.

[SRTP] Baugher, McGrew, Oran и др: *Протокол защищенной передачи данных в режиме реального времени*; draft-ietf-avt-srtp-07.txt, RFC xxxx; *Специальная группа инженерной поддержки сети Интернет*, 2003 г.

{Примечание издателя: Этот стандарт (RFC#) будет включен, когда он появится}

[TLS] DIEKS (T.), ALLEN (C.): *Протокол TLS версии 1.0, RFC 2246, Специальная группа инженерной поддержки сети Интернет*, 1999 г.

СЕРИИ РЕКОМЕНДАЦИЙ МСЭ-Т

Серия A	Организация работы МСЭ-Т
Серия B	Средства выражения: определения, символы, классификация
Серия C	Общая статистика электросвязи
Серия D	Общие принципы тарификации
Серия E	Работа сети в целом, служба телефонной связи, работа служб и человеческий фактор
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 телевизионных, звуковых программ и других мультимедийных сигналов
Серия K	Защита от помех
Серия L	Конструкция, установка и защита кабелей и других элементов линейно-кабельных сооружений
Серия M	TMN и техническое обслуживание сетей: международные системы передачи, телефонные, телеграфные, факсимильные и арендованные каналы
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 и взаимосвязь открытых систем
Серия Y	Глобальная информационная инфраструктура, аспекты межсетевого протокола (IP) и сети следующих поколений
Серия Z	Языки и общие аспекты программного обеспечения для систем электросвязи

Отпечатано в Швейцарии
Женева, 2005 г.