МЕЖДУНАРОДНЫЙ СОЮЗ ЭЛЕКТРОСВЯЗИ

# МСЭ-Т H.235.8

СЕКТОР СТАНДАРТИЗАЦИИ
ЭЛЕКТРОСВЯЗИ МСЭ

(09/2005)

Серия H: АУДИОВИЗУАЛЬНЫЕ И
МУЛЬТИМЕДИЙНЫЕ СИСТЕМЫ

Инфраструктура аудиовизуальных услуг –
Системные аспекты

## Безопасность H.323: Обмен ключами для SRTP с использованием защищенных каналов сигнализации

Рекомендация МСЭ-Т H.235.8

РЕКОМЕНДАЦИИ МСЭ-Т СЕРИИ H

**АУДИОВИЗУАЛЬНЫЕ И МУЛЬТИМЕДИЙНЫЕ СИСТЕМЫ**

| | |
|---|---|
| ХАРАКТЕРИСТИКИ ВИДЕОТЕЛЕФОННЫХ СИСТЕМ | H.100–H.199 |
| ИНФРАСТРУКТУРА АУДИОВИЗУАЛЬНЫХ УСЛУГ | |
| 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 движущихся видеоизображений | H.260–H.279 |
| Сопутствующие системные аспекты | H.280–H.299 |
| Системы и оконечное оборудование для аудиовизуальных услуг | H.300–H.349 |
| Архитектура услуг справочника для аудиовизуальных и мультимедийных услуг | H.350–H.359 |
| Качество архитектуры обслуживания для аудиовизуальных и мультимедийных услуг | H.360–H.369 |
| Дополнительные услуги для мультимедиа | 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 услуг серии H | H.510–H.519 |
| Приложения и услуги мобильной мультимедийной совместной работы | H.520–H.529 |
| Защита мобильных мультимедийных систем и услуг | H.530–H.539 |
| Защита приложений и услуг 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 И МУЛЬТИМЕДИЙНЫЕ TRIPLE-PLAY УСЛУГИ | |
| 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.235.8

## Безопасность H.323: Обмен ключами для SRTP с использованием защищенных каналов сигнализации

**Резюме**

Целью данной Рекомендации является описание процедур защиты для обмена ключами для SRTP с использованием защищенных каналов сигнализации в сетях H.323/H.235.

Данную Рекомендацию следует использовать совместно с Рекомендациями МСЭ-Т H.323 и H.225.0 версии 4 и далее.

**Источник**

Рекомендация МСЭ-Т H.235.8 утверждена 13 сентября 2005 года 16-й Исследовательской комиссией МСЭ-Т (2005–2008 гг.) в соответствии с процедурой, изложенной в Рекомендации МСЭ-Т A.8.

## ПРЕДИСЛОВИЕ

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

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

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

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

## ПРИМЕЧАНИЕ

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

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

## ПРАВА ИНТЕЛЛЕКТУАЛЬНОЙ СОБСТВЕННОСТИ

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

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

## СОДЕРЖАНИЕ

# Рекомендация МСЭ-Т H.235.8

## Безопасность H.323: Обмен ключами для SRTP с использованием защищенных каналов сигнализации

### 1 Сфера применения

Целью данной Рекомендации является описание процедур защиты для поддержки протокола защиты передачи данных в режиме реального времени (SRTP) IETF между конечными точками H.323 в случаях, когда криптографический материал для медиаканала передается по защищенным каналам сигнализации, например, IPsec (RFC 2401), TLS (RFC 2246) или другим алгоритмам H.235. Данные процедуры защиты предлагаются в качестве альтернативы другим процедурам защиты H.235, которые поддерживают SRTP.

В данной Рекомендации описываются процедуры, использовавшиеся для поддержки протокола SRTP IETF в Рек. МСЭ-Т H.323. SRTP обеспечивает защиту медиа-RTP и при обеспечении услуг по управлению ключами и согласованию криптографических параметров опирается на отдельные протоколы. Данные процедуры не следует использовать, когда защищенный канал сигнализации завершается на промежуточном устройстве, в таких случаях следует передавать криптографический материал при помощи сквозного механизма защиты.

Данные процедуры поддерживают сигнализацию, согласование и передачу криптографических ключей SRTP, идентификаторов алгоритмов аутентификации и шифрования и другие параметры сеанса между конечными точками H.323.

Ключевым аспектом данных процедур является то, что и подчиненное устройство H.245, и главное устройство H.245 должны быть в состоянии генерировать и распределять криптографические ключи.

Обмен возможностями защиты SRTP может происходить с использованием существующего обмена возможностями оконечных устройств с использованием записей h235SecurityCapability в capabilityTable сообщения TerminalCapabilitySet H.245. Поле genericH235SecurityCapability в поле encryptionAuthenticationAndIntegrity в записи h235SecurityCapability содержит поле SrtpCryptoCapability, в котором описываются криптоблоки SRTP.

Криптопараметр SRTP устанавливается для сигнализации и согласования криптографических параметров SRTP. Под криптопараметром в данной Рекомендации подразумеваются двусторонние одноадресные медиапотоки, где каждый источник обладает уникальным криптографическим ключом; поддержка многоадресных медиапотоков или многоточечных одноадресных потоков является предметом дальнейшего изучения.

Предполагается, что криптопараметр SRTP способен установить криптографические параметры SRTP в одном сообщении или за один проход сообщений туда и обратно. В случае обмена сообщениями за один проход туда и обратно, могут быть согласованы криптографические параметры. Например, в Fast Connect, вызывающая конечная точка H.323 посылает набор предлагаемых криптопараметров SRTP отвечающей конечной точке Endpoint, при этом каждое предложение инкапсулировано в отдельное сообщение H.245 OpenLogicalChannel. Затем, отвечающая конечная точка H.323 может принять один из предлагаемых параметров и послать ответ, который включает в себя выбранное подмножество параметров, инкапсулированное в сообщение H.245 OpenLogicalChannel.

В случае однократного обмена сообщениями согласования не происходит. Вызывающая конечная точка H.323 посылает криптопараметры отвечающей конечной точке H.323, которая либо принимает предложенные параметры, либо отказывает в соединении.

Для обеспечения сквозной конфиденциальности и аутентификации материала сеансовых ключей SRTP, которыми обмениваются конечные точки H.323, могут быть добавлены процедуры шифрования открытым ключом. Это осуществляется путем шифрования и подписи материала ключей SRTP в случае, когда инкапсулирующий протокол безопасности, например, IPsec, TLS, завершается на промежуточном устройстве, и поэтому сквозная безопасность не обеспечивается.

## 2 Справочные документы

### 2.1 Нормативные справочные документы

В перечисленных ниже Рекомендациях МСЭ-Т и другой справочной литературе содержатся положения, которые посредством ссылок на них в этом тексте составляют основные положения данной Рекомендации. На момент опубликования, действовали указанные редакции документов. Все Рекомендации и другая справочная литература, являются предметом корректировки, и стороны пришли к договоренности основываться на этой Рекомендации и стараться изыскивать возможность для использования самых последних изданий Рекомендации и справочной литературы, перечисленной ниже. Регулярно публикуется перечень действующих Рекомендаций МСЭ-Т. Ссылка на документ в рамках этой Рекомендации не дает ему, как отдельному документу, статуса Рекомендации.

– ITU-T Recommendation H.225.0 (2003 г.), *Протоколы сигнализации о соединении и пакетирование потоков носителей для мультимедийных систем связи на основе пакетов*.

– ITU-T Recommendation H.235.0 (2005), *H.323 security: Framework for security in H-series (H.323 and other H.245-based) multimedia systems*.

– ITU-T Recommendation H.323 (2003 г.), *Мультимедийные системы связи на основе пакетов*.

– ITU-T Recommendation H.460.11 (2004 г.), *Отсроченное установление вызова в системах H.323*.

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

– IETF RFC 2401 (1998), *Security Architecture for the Internet Protocol*.

– IETF RFC 2733 (1999), *An RTP Payload Format for Generic Forward Error Correction*.

– IETF RFC 3280 (2002), *Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile*.

– IETF RFC 3550 (2003), *RTP: A Transport Protocol for Real-Time Applications*.

– IETF RFC 3711 (2004), *The Secure Real-time Transport Protocol (SRTP)*.

– IETF RFC 3852 (2004), *Cryptographic Message Syntax (CMS)*.

### 2.2 Информативные справочные документы

– IETF Draft, F. Andreasen, M. Baugher, D. Wing: *Session Description Protocol Security Descriptions for Media Streams,* <draft-ietf-mmusic-sdescriptions-11.txt>.

## 3 Символы и сокращения

В данной Рекомендации используются следующие сокращения:

| | | |
|---|---|---|
| AES | Advanced Encryption Algorithm | Улучшенный стандарт шифрования, стандарт AES |
| ASN.1 | Abstract Syntax Notation One | Абстрактно-синтаксическая нотация версии 1 |
| CA | Certificate Authority | Органы сертификации |
| CEK | Content Encryption Key | Ключ шифрования содержимого |
| CMS | Cryptographic Message Syntax | Синтаксис криптографических сообщений |
| EP | Endpoint | Конечная точка |
| FEC | Forward Error Correction | Упреждающая коррекция ошибок |
| FFS | For Further Study | Для дальнейшего изучения |
| F8 | Encryption Algorithm | Алгоритм шифрования UMTS |
| GK | Gatekeeper | Привратник |

| | | |
|---|---|---|
| GW | Gateway | Шлюз |
| HMAC | Keyed-Hash Message Authentication Code | Код аутентификации сообщения, использующий хеш-функцию |
| IETF | Internet Engineering Task Force | Комитет по инженерным вопросам интернета, Комитет IETF |
| KDR | Key Derivation Rate | Скорость вычисления ключа |
| MAC | Message Authentication Code | Код аутентификации сообщения |
| MKI | Master Key Identifier | Идентификатор главного ключа |
| OID | Object Identifier | Идентификатор объекта |
| OLC | Open Logical Channel | Открытый логический канал |
| PKI | Public Key Infrastructure | Инфраструктура открытых ключей |
| RAS | Registration, Admission, Status | Регистрация, допуск, статус |
| ROC | Roll-over Counter | Автоматически переводящийся счетчик |
| RTCP | Real-time Transport Control Protocol | Протокол контроля транспорта в режиме реального времени |
| RTP | Real-time Transport Protocol | Протокол транспорта в режиме реального времени |
| SHA1 | Secure Hash Algorithm 1 | Алгоритм аутентификации и проверки целостности информации версии 1, алгоритм SHA1 |
| SRTCP | Secure Real-time Transport Control Protocol | Протокол защиты контроля транспорта в режиме реального времени |
| SRTP | Secure Real-time Transport Protocol | Протокол защиты транспорта в режиме реального времени |
| SSRC | Synchronization Source | Источник синхронизации |
| TLS | Transport Level Security | Защита транспортного уровня |
| WSH | Window Size Hint | Рекомендуемый размер окна |

## 4 Описание параметров

Обмен криптографическими возможностями и материалом ключей SRTP осуществляется с использованием двух параметров:

– **SrtpCryptoInfo** внутри **StrpCryptoCapability** должен содержать криптоблок и параметры сеанса. Для сигнализации и согласования криптографических параметров SRTP следует передавать параметр **SrtpCryptoInfo** в параметре H.245 **genericH235SecurityCapability**.

– **SrtpKeyParameters** внутри **SrtpKeys** должен содержать материал ключа SRTP. Контейнер **SrtpKeys** в параметре H.245 **h235Key** должен осуществлять передачу одного или более **SrtpKeyParameters** с ключами SRTP.

Под криптопараметром SRTP в данной Рекомендации подразумеваются двусторонние одноадресные медиапотоки, где каждый источник обладает уникальным криптографическим ключом; поддержка многоадресных медиапотоков или многоточечных одноадресных потоков является предметом дальнейшего изучения.

### 4.1 Параметр передачи SRTP

Дуплексное соединение медиа-SRTP состоит из двух однонаправленных каналов по одному в каждом направлении. Каждое криптопредложение передается в отдельном сообщении **OpenLogicalChannel** H.245.

### 4.1.1 Передача SrtpKeys

Следует передавать материал криптографического ключа SRTP **SrtpKeys** в поле **genericKeyMaterial** параметра **secureSharedSecret** (**V3KeySyncMaterial**), содержащегося в контейнере **h235Key** параметра **encryptionSync** сообщений **OpenLogicalChannel** H.245.

Содержание криптографического ключа SRTP в контейнере **genericKeyMaterial** следует идентифицировать с использованием значения идентификатора объекта H.235.8 (см. таблицу 1) в поле **standard** поля **capabilityIdentifier** внутри поля **genericH235SecurityCapability** поля **encryptionAuthenticationAndIntegrity** в **h235Media dataType** OLC.

Альтернативные предложения **OpenLogicalChannel** для того же канала, содержащие такое же значение **sessionID** в **H2250LogicalChannelParameters** могут использовать то же самое криптопредложение. Поскольку принят будет только один из этих альтернативных сеансов, уникальность ключа будет гарантирована.

### 4.1.2 Передача SrtpCryptoCapability

Передавать параметр **SrtpCryptoCapability** следует в поле **genericH235SecurityCapability** поля **encryptionAuthenticationAndIntegrity** в **h235Media** параметра **dataType** сообщений **OpenLogicalChannel**.

Сообщение H.245 **TerminalCapabilitySet** может включать в себя одну или более записей **h235SecurityCapability** в **capabilityTable**. Для того чтобы обозначить поддержку для данных процедур, конечная точка H.323 должна установить следующие значения поля **genericH235SecurityCapability** внутри **encryptionAuthenticationAndIntegrity** в записи **h235SecurityCapability**:

- **capabilityIdentifier** должно содержать OID H.235.8 (см. таблицу 1) в поле **standard**;
- **maxbitRate**, **collapsing**, **nonCollapsing**, и **transport** не используются;
- **nonCollapsingRaw** должно содержать параметр **SrtpCryptoCapability**.

**Таблица 1/H.235.8 – Идентификатор объекта H.235.8**

| Значение OID |
|---|
| { itu-t (0) recommendation (0) h (8) 235 version (0) 4 90 } |

## 4.2 Описание параметра SrtpCryptoCapability

Параметр **SrtpCryptoCapability** может содержать один или более параметров **SrtpCryptoInfo**, которые могут быть использованы для описания возможностей для сеанса SRTP. Элементы **BOOLEAN OPTIONAL** следует понимать следующим образом:

1) если FALSE, возможность не поддерживается;

2) если TRUE, возможность поддерживается и является обязательной;

3) если отсутствует, возможность поддерживается, но не является обязательной.

При использовании **SrtpCryptoCapability** при обмене возможностями, можно обозначить все приемлемые опции в одной общей возможности. При таком использовании пропуск элемента **BOOLEAN OPTIONAL** будет означать, что возможность поддерживается, но не является обязательной.

При использовании в выражении OLC **dataType** может использоваться только одна опция. При этом следует соблюдать следующие правила:

- **FecOrder** может содержать только одно из дополнительных значений.
- В **SrtpSessionParameters**, значения **BOOLEAN OPTIONAL** должны быть только TRUE или FALSE.
- **SrtpCryptoCapability** должно содержать только один элемент **SrtpCryptoInfo**.

Параметр **SrtpCryptoInfo** состоит из обязательного для заполнения поля **cryptoSuite** и дополнительных полей **sessionParams** и **allowMKI,** описание которых приводится ниже.

**Таблица 2/H.235.8 – Идентификаторы объекта криптоблока H.235.8**

| Криптоблок | Значение OID |
|---|---|
| AES_CM_128_HMAC_SHA1_80 | { itu-t (0) recommendation (0) h (8) 235 version (0) 4 91 } |
| AES_CM_128_HMAC_SHA1_32 | { itu-t (0) recommendation (0) h (8) 235 version (0) 4 92 } |
| F8_128_HMAC_SHA1_80 | { itu-t (0) recommendation (0) h (8) 235 version (0) 4 93 } |

### 4.2.1 cryptoSuite

Идентификатор объекта (см. таблицу 2) в поле **cryptoSuite** обозначает алгоритмы шифрования и аутентификации, необходимые для использования в сеанса SRTP.

Многочисленные параметры спецификации SRTP объединены в три опции, называемые "Криптоблоки". Они являются расширяемыми, так как можно добавлять новые Криптоблоки. AES_CM_128_HMAC_SHA1_80, AES_CM_128_HMAC_SHA1_32, и F8_128_HMAC_SHA1_80 являются определенными криптоблоками. Параметры SRTP, которые объединяются в криптоблоки, приведены в таблице 3.

**Таблица 3/H.235.8 – Заданные по умолчанию значения криптоблоков**

| SRTP parameter | AES_CM_128_ HMAC_SHA1_80 | AES_CM_128_ HMAC_SHA1_32 | F8_128_ HMAC_SHA1_80 |
|---|---|---|---|
| Длина главного ключа | 128 битов | 128 битов | 128 битов |
| Значение расширения (Salt) | 112 битов | 112 битов | 112 битов |
| Время жизни | 2^31 пакетов | 2^31 пакетов | 2^31 пакетов |
| Шифр | AES Counter | AES Counter | F8 |
| Ключ шифрования | 128 битов | 128 битов | 128 битов |
| MAC | HMAC-SHA1 | HMAC-SHA1 | HMAC-SHA1 |
| Длина метки аутентификации | 80 битов | 32 бита | 80 битов |
| Аут. SRTP длина ключа | 160 битов | 160 битов | 160 битов |
| Аут. SRTCP длина ключа | 160битов | 160 битов | 160 битов |

Поле **cryptoSuite** является параметром, подлежащим согласованию.

### 4.2.2 sessionParams

Параметры сеанса могут быть либо подлежащими согласованию, либо декларативными; в определении отдельного параметра сеанса должно указываться, является ли он согласуемым или декларативным. Согласуемые параметры применяются к данным, пересылаемым в обоих направлениях, а декларативные параметры применяются только к медиаданным, посылаемым объектом, который генерировал описание сеанса. Таким образом, декларативный параметр в предложении применяется к медиаданным, посылаемым тем, кто делает предложение, а декларативный параметр в ответе применяется к медиаданным, посылаемым отвечающим.

Дополнительное поле **sessionParams** содержит параметры сеанса SRTP.

#### 4.2.2.1 kdr

KDR описывает скорость образования ключа, как описано в пункте 4.3.1 RFC 3711. Значение должно представлять собой целое число в множестве {1, 2,..., 24}, которое обозначает степень 2 от 2^1 до 2^24, включительно. Уровень образования ключа SRTP контролирует то, как часто образуется новый сеансовый ключ из главного ключа SRTP (RFC 3711). Когда скорость образования ключа не определена (например, параметр KDR пропущен), происходит однократное первоначальное образование ключа (RFC 3711). KDR является декларативным параметром.

#### 4.2.2.2 unencryptedSrtp

Это дополнительное поле Boolean: если данное поле присутствует, оно сигнализирует, что полезные нагрузки пакета SRTP не шифруются. Это согласуемый параметр.

#### 4.2.2.3 unencryptedSrtcp

Это дополнительное поле Boolean: если данное поле присутствует, оно сигнализирует, что полезные нагрузки пакета SRTCP не шифруются. Это согласуемый параметр.

#### 4.2.2.4 unauthenticatedSrtp

Пакеты SRTP и SRTCP аутентифицируются по умолчанию. Это дополнительное поле Boolean: если данное поле присутствует, оно сигнализирует, что полезные нагрузки пакета SRTP не аутентифицируются. Спецификации SRTP требуют использования аутентификации сообщения для SRTCP, но не для SRTP (RFC 3711) Это согласуемый параметр.

#### 4.2.2.5 fecOrder

**fecOrder** сигнализирует порядок обработки упреждающей коррекции ошибок для пакетов RTP (RFC 3550, RFC 2733), соответствующий шифрованию SRTP у отправителя. Значение для **fecOrder** поля **fecBeforeSrtp** сигнализирует, что FEC применяется до обработки отправителем SRTP медиа-SRTP и после обработки получателем SRTP медиа-SRTP; **fecBeforeSrtp** задается по умолчанию. Обработка **fecAfterSrtp** осуществляется в обратном порядке. **fecOrder** является декларативным параметром.

#### 4.2.2.6 windowSizeHint

SRTP определяет параметр SRTP-WINDOW-SIZE (пункт 3.3.2 RFC 3711) для защиты от атак замещения оригинала. Минимальное значение – 64 (RFC 3711), однако, это значение может оказаться слишком низким для некоторых приложений, например, видео.

Параметр сеанса Window Size Hint (WSH) содержит указание на то, насколько велико должно быть это окно для удовлетворительной работы (например, на основе знания отправителя о количестве пакетов в секунду). Однако, для удовлетворительного образования получателем параметра может быть достаточно информации, данной в дескрипторах медиапакетизации. Поэтому, данное значение считается лишь указанием для получателя, который может проигнорировать предоставленное значение.

**windowSizeHint** является декларативным параметром.

#### 4.2.2.7 Определение новых параметров сеанса SRTP

Новые параметры сеанса SRTP являются обязательными по умолчанию. Поле **newParameter** используется в качестве механизма расширения для новых параметров сеанса. Если старая конечная точка H.323 получает параметр **SrtpCryptoInfo** с неизвестным параметром сеанса в поле **newParameter**, этот новый параметр **SrtpCryptoInfo** следует считать ошибочным.

### 4.3 Описание параметра SrtpKeys

Поле **SrtpKeys** содержит один или более ключевых параметров **SrtpKeyParameter**, которые следует использовать для сеанса SRTP. Каждый **SrtpKeyParameter** содержит материал ключа (главный ключ и расширение) и все правила, относящиеся к данному главному ключу, включая то, как долго его можно использовать (время жизни) и использует или не использует идентификатор главного ключа (MKI) для ассоциирования входящего пакета SRTP с определенным главным ключом. Совместимые реализации согласуются с правилами, связанными с главным ключом, и не должны принимать входящие пакеты, которые нарушают правила (например, после того, как время жизни главного ключа истекло).

#### 4.3.1 masterKey

Это криптографический главный ключ, который следует использовать для сеанса SRTP. Длина ключа определяется криптоблоком, для которого применяется ключ. Если длина ключа не совпадает с длиной ключа в криптоблоке, следует считать данный криптопараметр неверным. Каждый главный

ключ должен представлять собой криптографически случайное число и должен быть уникальным для предлагаемого медиапотока.

### 4.3.2 masterSalt

Это криптографическое главное расширение, которое следует использовать для сеанса SRTP. Длина расширения определяется криптоблоком, для которого применяется ключ. Если длина расширения не совпадает с длиной расширения в криптоблоке, следует считать данный криптопараметр неверным. Каждое главное расширение должно представлять собой криптографически случайное число и должно быть уникальным для предлагаемого медиапотока.

### 4.3.3 lifetime

Данное поле обозначает дополнительное время жизни главного ключа, которое измеряется как максимальное число пакетов SRTP или SRTCP, использующих данный главный ключ (т. е. число пакетов SRTP и число пакетов SRTCP по отдельности должно быть меньше, чем время жизни). Значение времени жизни должно быть записано как целое положительное число, неравное нулю или как степень числа 2. Значение "времени жизни" не должно превышать максимальное время жизни пакета для криптоблока. Если время жизни слишком велико, или иначе – неверно, тогда неверным считается весь криптопараметр. Если поле времени жизни отсутствует, следует использовать заданное по умолчанию время жизни. Это удобно, когда время жизни криптографического ключа SRTP является значением, заданным по умолчанию.

### 4.3.4 masterKeyId

В данном дополнительном поле описываются правила, как следует идентифицировать ключи для сеанса SRTP. MKI является идентификатором главного ключа, связанного с главным ключом SRTP. Если дан MKI, необходимо также задать длину MKI. Длина MKI – это размер поля MKI в пакете SRTP, выраженный в байтах. Если длина MKI не задана, или ее значение превышает 128 (байтов), весь криптопараметр следует считать неверным.

Как отмечалось ранее, ключевой параметр может содержать один или более главных ключей. Когда ключевой параметр содержит более одного главного ключа, все главные ключи в данном ключевом параметре должны включать значение MKI. При использовании MKI, длина MKI должна быть одной и той же для всех ключей в данном криптопараметре.

## 4.4 Инициализация криптоконтекста SRTP

В добавление к различным параметрам SRTP, определенным выше, существует три аспекта, которые являются важными для функционирования заданных по умолчанию шифров SRTP:

- SSRC: Источник синхронизации;
- ROC: Автоматически переводящийся счетчик для заданного SSRC;
- SEQ: Порядковое число для заданного SSRC.

В одноадресном сеансе, как определено здесь, есть три ограничения этих значений. Первое ограничение – на SSRC, которое делает ключевой поток SRTP уникальным среди других участников. Как объясняется в SRTP, ключевой поток не следует использовать повторно на двух или более отрезках открытого текста.

Повторное использование ключевого потока делает зашифрованный текст уязвимым для криптоанализа. Одним из слабых мест является то, что известные поля открытого текста в одном потоке могут подвергнуть опасности участки повторно использованного ключевого потока, что далее может подвергнуть опасности большие участки открытого текста в других потоках. Поскольку ключевые потоки используют все современные шифровальные трансформации SRTP, совместное использование ключей представляет всеобщую проблему (RFC 3711). SRTP смягчает эту проблему путем включения SSRC отправителя в ключевой поток. Но SRTP не решает данную проблему полностью, потому что в Протоколе транспорта в режиме реального времени случаются конфликты SSRC, что бывает очень редко (RFC 3550), но все же возможно. Во время конфликта, два или более SSRC, которые разделяют главный ключ, будут иметь идентичные ключевые потоки для частично совпадающих участков пространства последовательного числа RTP. Описание защиты SRTP избегает повторного использования ключевого потока путем создания уникальных главных ключей, требующихся для отправителя и получателя. Таким образом, первое ограничение снимается.

Следует также заметить, что конфликты SSRC представляют еще одну проблему. SSRC используется для идентификации криптоконтекста и таким образом шифра, ключа ROC и т. д., для обработки входящих пакетов. В случае конфликтов SSRC идентификация криптоконтекста становится неоднозначной и правильной обработки пакета не происходит. Кроме того, если необходимо послать пакет RTCP BYE конфликтующему SSRC, необходимо также обеспечить защиту этого пакета.

Вторым ограничением является то, что когда SSRC начинает отсылку пакетов, значение ROC должно быть равно нулю. Таким образом, в описаниях защиты SRTP, которые ограничены одноадресными и парными, нет понятия "позднее присоединившийся" ("late joiner"). ROC и SEQ образуют "индекс пакета" в заданных по умолчанию преобразованиях SRTP (добавления, удаления и изменения имен доменов, присоединяемые к входящим и выходящим сообщениям) и, согласно данной Рекомендации, при начале сеанса значение ROC обычно устанавливается на ноль.

Третье ограничение заключается в том, что первоначальное значение SEQ должно выбираться из диапазона 0..2^15 – 1; это позволяет избежать неопределенности, если пакеты теряются в начале сеанса. Если в начале сеанса источник SSRC мог случайным образом выбрать большое значение порядкового числа и создать для получателя неоднозначную ситуацию: если первоначальные пакеты теряются при передаче до точки, когда порядковое число выходит за пределы (т. е. превышает 2^16 – 1), получатель может не осознать, что необходимо увеличить его ROC. Ограничив первоначальное SEQ диапазоном 0..2^15 – 1, вычислением "индекса пакета" SRTP будет найдено правильное значение ROC, если только все первые пакеты 2^15 не потеряны (что представляется маловероятным, если вообще возможным). См. пункт 3.3.1 спецификации SRTP, касающийся вычисления "индекс пакета" (RFC 3771).

### 4.4.1 Динамическое связывание SSRC с криптоконтекстом

Индекс пакета зависит от SSRC, SEQ входящего пакета и ROC, который является переменной криптоконтекста SRTP. Таким образом, безопасность SRTP в большой степени зависит от уникальности SSRC. Принимая во внимание вышеупомянутые ограничения, можно создать одноадресные криптоблоки SRTP без необходимости согласования значений SSRC в описании безопасности SRTP. Вместо этого данной Рекомендацией рекомендуется подход, называемый "динамическое связывание" ("late binding"). Когда приходит пакет, содержащийся в нем SSRC можно связать с криптоконтекстом во время начала сеанса (т. е. прибытие пакета SRTP), а не во время сигнализации сеанса (т. е. получения сообщения H.245). С прибытием пакета, содержащего SSRC, все элементы данных, необходимые для криптоконтекста SRTP, находятся у получателя (заметьте, что значение ROC по определению равно нулю; если требуется поддержка значений неравных нулю, потребуется дополнительная сигнализация). Другими словами, криптоконтекст для защищенного сеанса RTP, использующей динамическое связывание, первоначально идентифицируется сообщением H.245 как:

<*, address, port>

где '*' является групповым символом SSRC, "address" – локальный адрес получения из **mediaChannel**, а "port" – локальный порт получения из **portNumber**. Когда приходит первый пакет с ssrcX в поле SSRC, криптоконтекст

<ssrcX, address, port>

является приписываемым субъектом (instantiated subject) к следующим ограничениям:

- медиапакеты аутентифицируются: аутентификация должна пройти успешно; в противном случае, криптоконтекст не приписывается;
- медиапакеты не аутентифицируются: криптоконтекст приписывается автоматически.

Следует заметить, что использование динамического связывания, когда аутентификации пакетов медиа-SRTP не производится, подвержено атакам на безопасность и, следовательно, не рекомендуется (конечно, то же самое можно сказать о неаутентифицируемом SRTP в общем).

Заметьте, что использование динамического связывания без аутентификации приведет к созданию локального связывания (local state) как результата получения пакета от любого неизвестного SSRC. Поэтому использование неаутентифицированного SRTP не рекомендуется, так как оно провоцирует атаки типа "отказ в обслуживании". Динамическое связывание с аутентификацией, напротив, не создает подобных слабых мест в безопасности.

### 4.4.2 Разделение криптоконтекстов между сеансами или SSRC

Принимая во внимание ограничения и процедуры, описанные выше, нет необходимости явно сигнализировать SSRC, ROC и SEQ для одноадресного сеанса RTP. Поэтому для сигнализации SSRC, ROC или SEQ нет криптопараметров SRTP. Таким образом многочисленные SSRC из одного и того же объекта будут разделять криптопараметры SRTP, когда используется динамическое связывание. Многочисленные SSRC из одного и того же объекта возникают либо в результате использования многочисленных источников (микрофонов, камер и т. д.), либо использования полезных нагрузок RTP, требующих мультиплексирования SSRC в рамках одного и того же сеанса.

H.245 позволяет определять многочисленные сеансы RTP в одном и том же описании медиа, эти сеансы RTP будут также разделять криптопараметры SRTP. Приложение, которое использует криптопараметр SRTP таким образом разделяет главный ключ между сеансами RTP или SSRC и должно заменять главный ключ когда совокупность числа пакетов между всеми SSRC достигнет 2^31 пакетов. Источники SSRC, которые разделяют главный ключ должны быть уникальными.

Время жизни всех ключей, образованных из главного ключа, определяется временем жизни этого главного ключа. Таким образом, если время жизни главного ключа составляет 2^31 пакетов, и один из образованных ключей отослал 2^31 – y пакетов, тогда только y пакетов может быть отослано любым ключом, образованным от главного ключа. Это объясняется тем, что время жизни основывается на величине непредсказуемой части сигнала или случайности в ключе. При образовании ключа от главного ключа случайности не вводится, главный ключ сам является целиком случайностью или энтропией.

### 4.4.3 Удаление криптоконтекстов

Механизм, описанный выше, касается вопроса создания криптоконтекстов. Однако в действительности участники сеанса могут пожелать удалить криптоконтексты до окончания сеанса. Поскольку криптоконтекст содержит информацию, которую невозможно восстановить автоматически (например, ROC), важно, чтобы отправитель и получатель договорились о том, когда можно удалять криптоконтекст, и, что более важно, когда нельзя.

Даже если для одноадресного потока используется динамическое связывание, при удалении криптоконтекста ROC теряется и не может быть восстановлен автоматически (если значение не равно нулю).

Криптоконтексты должны быть удалены после получения **CloseLogicalChannel**. Кроме того, удаление криптоконтекста должно происходить согласно правилам удаления SSRC из таблицы членов (RFC 3711); заметьте, что удаление может стать результатом пакета SRTCP BYE или простоя из-за неактивности. Неактивные участники сеанса, желающие обеспечить "сохранность" своих криптоконтекстов должны посылать пакеты SRTCP через регулярные промежутки времени.

## 5 Процедуры

Следует использовать описанные ниже процедуры SRTP только для согласования защиты для одноадресных медиапотоков с двумя участниками в ситуациях, когда защита канала сигнализации H.245 обеспечивается путем инкапсулирования протокола защиты данных, например, IPsec (RFC 2401), TLS (RFC 2246). Обмен криптопараметрами SRTP с использованием сообщений H.245 должен выполнять следующие функции:

1) обмен и согласование возможностей шифрования и целостности медиа-SRTP.
2) согласование и создание первоначального шифрования и алгоритмов, ключей и параметров сеанса для использования для потоков SRTP в каждом направлении.
3) модификация шифрования и алгоритмов, ключей и параметров сеанса в любой момент сеанса SRTP.

### 5.1 Обмен возможностями защиты

Идентификация криптоблоков SRTP, алгоритмов шифрования и целостности, которые может поддерживать конечная точка H.323 должна проводиться **SrtpCryptoCapability**.

Обмен возможностями защиты будет осуществляться посредством существующего обмена возможностями оконечного устройства (существующего оконечного устройства) с использованием одной или более записей **h235SecurityCapability** в **capabilityTable** сообщения H.245 **TerminalCapabilitySet**. Поле **mediaCapability** в записи **h235SecurityCapability** поля **capabilityTable** используется для связывания возможности безопасности с определенной записью возможности медиа в **capabilityTable**.

Поле **encryptionAuthenticationAndIntegrity** в записи **h235SecurityCapability** содержит поле **genericH235SecurityCapability**, в котором определяются криптоблоки SRTP, идентифицируемые идентификаторами OID H.235.8. Если поле **standard** поля **capabilityIdentifier** поля **genericH235SecurityCapability** содержит OID H.235.8 (см. таблицу 1), **SrtpCryptoCapability** будет содержать один или более параметров **SrtpCryptoInfo,** которые обозначают криптоблоки, которые поддерживает конечная точка H.323. Поле **cryptoSuite** в поле **SrtpCryptoInfo** содержит OID как описано в таблице 2, которая идентифицирует определенный криптоблок. Внутри поля **SrtpCryptoInfo** в поле **sessionParams** идентифицируются параметры сеанса, а в поле **allowMKI** отображается, поддерживает ли конечная точка H.323 идентификатор MKI.

## 5.2 Первоначальное согласование

### 5.2.1 Первоначальное предложение

Каждое криптопредложение передается в отдельном сообщении **OpenLogicalChannel**. Каждое криптопредложение должно содержать одну структуру **SrtpCryptoInfo** в поле **SrtpCryptoCapability** и одну или более структур **SrtpKeyParameters** в поле **SrtpKeys**.

Для обычных процедур H.245 (не быстрое соединение) конечная точка H.323 должна включать криптопредложение, как описано в структурах **SrtpCryptoInfo** и **SrtpKeyParameters** в сообщении **OpenLogicalChannel** H.245 для прямого направления (от предлагающей конечной точки H.323 к отвечающей конечной точке H.323). Конечная точка H.323 должна предложить наиболее предпочтительную возможность безопасности главного устройства, как было указано во время обмена возможностями оконечных устройств, и для которой она сама имеет возможность.

Для процедур быстрого соединения предлагающая конечная точка H.323 должна послать криптопредложение как описано в структурах **SrtpCryptoInfo** и **SrtpKeyParameters** в отдельных сообщениях **OpenLogicalChannel** для прямого направления (от предлагающей конечной точки H.323 к отвечающей конечной точке H.323).

Предлагаемые сообщения **OpenLogicalChannel** должны быть выстроены в порядки убывания предпочтительности, при этом наиболее предпочтительный криптоблок должен быть первым в списке, более предпочтительные криптоблоки должны быть криптографически более сильными, чем менее предпочтительные криптоблоки. В общем, более предпочтительные криптоблоки должны быть криптографически более сильными, чем менее предпочтительные криптоблоки.

Делая криптопредложение, предлагающий должен быть готов поддерживать защиту медиа в соответствии с любым из предложенных криптопараметров. В этой связи существует две проблемы. Во-первых, предлагающий не знает, какой ключ будет использовать отвечающий для медиаданных, посланных предлагающим. Поскольку медиаданные могут придти раньше криптоответа, может произойти задержка или ограничение. Если это является неприемлемым для предлагающего, он должен использовать механизм такой, например, как процедуры установления соединения с задержкой H.460.11 для предотвращения вышеупомянутой проблемы.

При большом количестве предложений может возникнуть другая проблема. Предлагающий не может определить, какое из предложений принял отвечающий, пока не получен криптоответ, кроме того медиа могут придти раньше криптоответа. Если это является неприемлемым для предлагающего, ему следует либо не посылать более одного предложения, либо использовать алгоритм такой, например, как процедуры установления соединения с задержкой H.460.11 для предотвращения вышеупомянутой проблемы.

**SrtpCryptoInfo** может включать параметры сеанса.

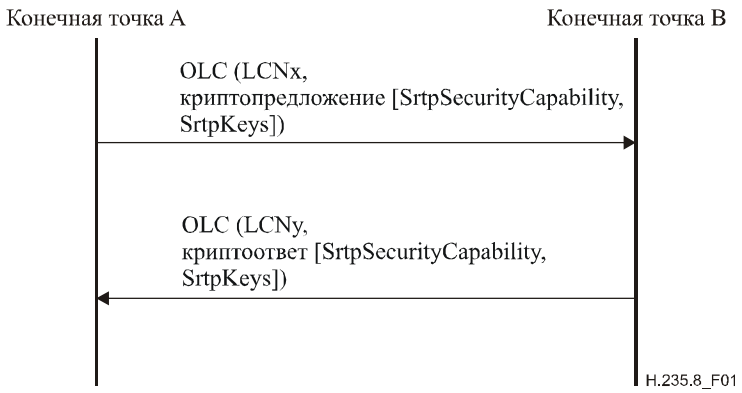


**Рисунок 1/H.235.8 – Обмен сообщениями предложение/ответ для быстрого соединения**

#### 5.2.1.1 Первоначальный криптоответ

##### 5.2.1.1.1 Общие сведения

Данные процедуры применимы для быстрого соединения (Fast Connect) и для обычных процедур H.245. Криптоответ должен содержать одну структуру **SrtpCryptoInfo** в **SrtpCryptoCapability** и одну или более структур **SrtpKeyParameters** в **SrtpKeys**.

Отвечающая конечная точка H.323 должна применить криптоблок, выбранный из криптопредложения, к соответствующему однонаправленному каналу SRTP в обратном направлении и должна сгенерировать ключ(и) для использования в данном канале SRTP в обратном направлении.

Кроме того, отвечающая конечная точка H.323 должна включить один или более ключей в **SrtpKeys**, которые следует использовать для потока SRTP от отвечающей конечной точки H.323 к предлагающей конечной точке H.323. Отвечающая конечная точка H.323 может также включить любые параметры сеанса из криптопредложения, которые она желает согласовать.

Акцептованы могут быть только действительные параметры; действительные параметры не нарушают никаких общих правил, определенных для описаний защиты, так же, как и любых особых правил, определенных для рассматриваемых передачи и метода ключа.

В случае быстрого соединения при выборе одного из действительных криптопредложений, отвечающий должен выбрать наиболее предпочтительное криптопредложение, которое он может поддерживать, т. е. первый действительный поддерживаемый параметр в списке, принимая во внимание возможности отвечающего и политику защиты. Если ни одно из предложений не является действительным, или ни одно из предложений не поддерживается, предложенный медиа поток должен быть отклонен.

Если криптопредложение принимается, криптоответ должен содержать ключ(и), который(ые) будет использовать отвечающий для медиа, посланных предлагающим. Заметьте, что ключ должен быть предоставлен, вне зависимости от параметров любого направления в предложении или ответе.

Кроме того, в криптоответ необходимо включить любые согласуемые параметры сеанса. Декларативные параметры сеанса, предоставляемые предлагающим, не включаются в криптоответ, однако отвечающий может предоставить свой собственный набор декларативных параметров сеанса.

После акцептования одного из предложенных криптопараметров, отвечающий может начать медиапересылку предлагающему в соответствии с выбранным криптопредложением. Однако заметьте, что пока не получен криптоответ, предлагающий может не смочь правильно обработать такие медиапакеты.

##### 5.2.1.1.2 Процедуры быстрого соединения

Для процедур быстрого соединения, отвечающая конечная точка H.323, получающая криптопредложения в одном или более сообщений **OpenLogicalChannel** H.245 должна принять одно из криптопредложений, послав сообщение **OpenLogicalChannel** H.245, содержащее криптоответ, как показано на рисунке 1, или отклонить все криптопредложения, послав **ReleaseComplete** с **ReleaseCompleteReason** со значением **securityDenied**, или послав элемент **FastConnectRefused** в

сообщении H.225.0. Если отвечающая конечная точка H.323 не поддерживает данную Рекомендацию или ни один из вариантов в криптопредложении, она должна отклонить криптопредложение, послав **ReleaseComplete** с **ReleaseCompleteReason** со значением **securityDenied**, или послав элемент **FastConnectRefused** в сообщении H.225.0.

#### 5.2.1.1.3 Обычные процедуры H.245

Для обычных процедур H.245 (не быстрое соединение) применяется следующая процедура. Если конечная точка H.323 еще не отослала **OpenLogicalChannel**, содержащее криптопредложение до получения **OpenLogicalChannel**, содержащего криптопредложение, она должна послать **OpenLogicalChannelAck**, а затем **OpenLogicalChannel**, содержащее криптоответ как показано на рисунке 2.

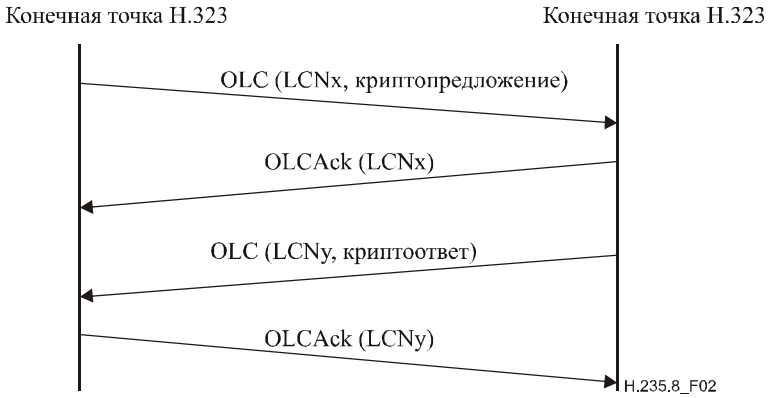


**Рисунок 2/H.235.8 – Обмен сообщениями предложение/ответ**

Если конечная точка H.323 уже послала **OpenLogicalChannel**, содержащее криптопредложение, до получения **OpenLogicalChannel**, содержащего криптопредложение, главные и подчиненные конечные точки H.323 должны предпринять следующие действия:

1) главная конечная точка H.323 должна обработать полученное криптопредложение и, если оно совместимо с криптопредложением, которое она уже отослала, она должна акцептовать полученное криптопредложение как криптоответ, отправив **OpenLogicalChannelAck**, как показано на рисунке 3. Если полученное криптопредложение не совместимо с криптопредложением, которое она уже отослала, она должна отклонить полученное криптопредложение, отправив **OpenLogicalChannelReject** со значением **cause** поля **securityDenied**, как показано на рисунке 4. Термин "совместимый" означает, что в криптопредложении и криптоответе должны совпадать следующие параметры: **cryptoSuite** и согласуемые параметры сеанса;

2) подчиненная конечная точка H.323 должна обработать полученное криптопредложение и, если оно совместимо с криптопредложением, которое она уже отослала, она должна акцептовать полученное криптопредложение как криптоответ, отправив **OpenLogicalChannelAck**, как показано на рисунке 3. Если полученное криптопредложение не совместимо с криптопредложением, которое она уже отослала, и, если она желает принять данное криптопредложение, она должна сделать это, послав следующие сообщения, показанные на рисунке 4.

   a) **OpenLogicalChannelAck** чтобы принять первоначальное криптопредложение от главного устройства (master);

   b) **CloseLogicalChannel** чтобы завершить свое собственное криптопредложение если от главного устройства (master) еще не было получено **OpenLogicalChannelReject**;

   c) **OpenLogicalChannel** с криптоответом, который совпадает с криптопредложением от главного устройства (master).

Если подчиненная конечная точка H.323 не поддерживает вариант в предложении или не желает акцептовать криптопредложение, она должна отклонить криптопредложение, послав **OpenLogicalChannelReject** со значением **securityDenied** в **cause**.

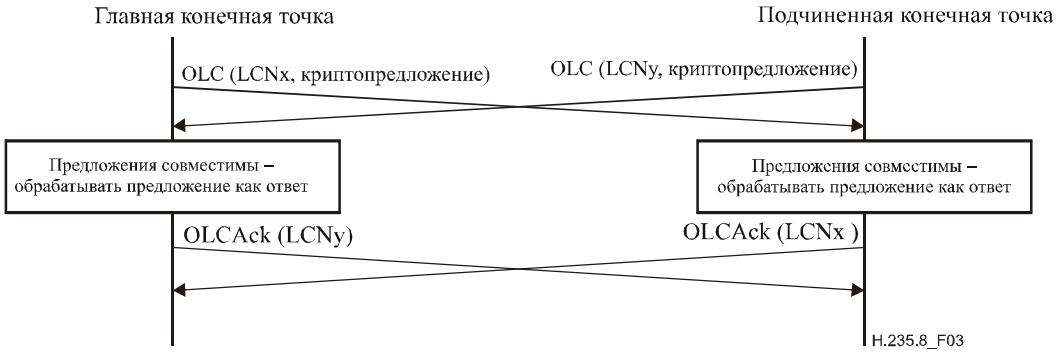


**Рисунок 3/H.235.8 – Одновременный совместимый обмен сообщениями предложение/ответ**

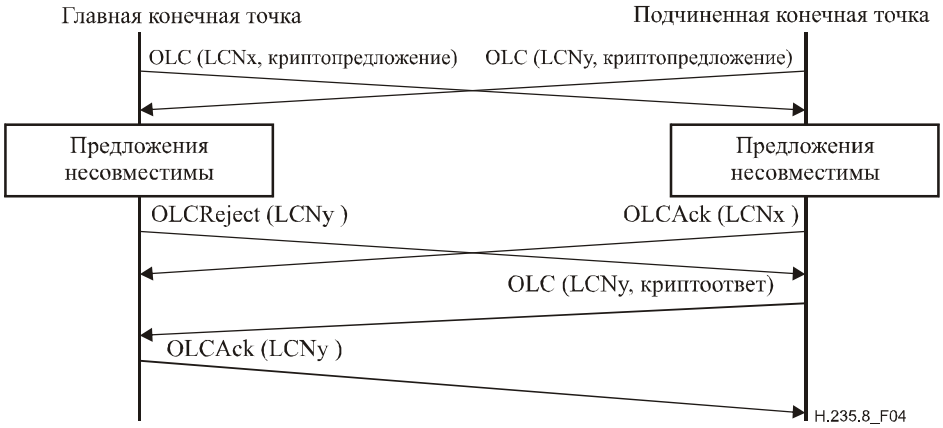


**Рисунок 4/H.235.8 – Одновременный несовместимый обмен сообщениями предложение/ответ**

#### 5.2.1.2 Обработка предлагающим первоначального ответа

Когда предлагающий получает криптоответ, он должен уточнить, какое из первоначальных криптопредложений было принято и продублировано в криптоответе. Также криптоответ должен содержать один или более ключей, которые будут использованы для медиа, пересылаемых от отвечающего предлагающему.

Предлагающий должен проверить, чтобы ключи в криптоответе не совпадали с ключами в криптопредложении. Если криптопредложение содержало какие либо обязательные для применения параметры сеанса, предлагающий должен проверить, что указанные параметры включены в криптоответ и совпадают с соответствующими параметрами в криптопредложении. Если криптоответ содержит какие-либо обязательные для применения декларативные параметры сеанса, предлагающий должен быть способен поддерживать их.

Если не удается выполнить какое-либо из указанных условий, следует считать, что согласование не удалось.

### 5.3 Модификация сеанса

После создания потока медиа-SRTP, можно модифицировать данный поток в любое время, используя новый обмен сообщениями предложение/ответ для осуществления смены ключей или криптоблока. Для открытия нового логического канала, который заменит существующий логический канал, следует передавать новое криптопредложение и новый криптоответ в параметрах **SrtpCryptoCapability** и **SrtpKeys** сообщения **OpenLogicalChannel** H.245 с использованием процедур **replacementFo**r. Предлагающая конечная точка H.323 должна содержать криптопредложения в одном или более сообщений **OpenLogicalChannel** H.245 внутри сообщения H.225.0.

Отвечающая конечная точка H.323, получающая криптопредложения должна ответить акцептованием одного из предложений, послав сообщение **OpenLogicalChannel** H.245 внутри сообщения H.225.0, или отклонением предложений сообщением **OpenLogicalChannelReject** со значением **securityDenied** в **cause**. Если криптопредложение отклоняется, действует один из предыдущих криптопараметров.

При создании нового главного ключа, будет временной интервал, в течение которого конечная точка H.323 должна получить медиа, зашифрованные согласно старому и новому обмену предложение/ответ. MKI из входящего пакета SRTP необходимо использовать для связывания этого пакета либо со старым либо с новым главным ключом. По этой причине если ожидается, что во время сеанса произойдет смена ключей, при которой не изменятся адреса источника/пункта назначения и порты, использование MKI является обязательным для того, чтобы получатель мог идентифицировать связываемый/ассоциируемый материал ключа во время обмена ключами.

### 5.4 Отсутствие согласования

В случае отсутствия согласования криптоблока, криптографического ключа или параметров сеанса, параметры безопасности для потока определяет отправитель. Поскольку не существует механизма согласования, отправитель должен включить точно одно криптопредложение, а получатель должен либо принять, либо отклонить его, послав **ReleaseComplete** с **ReleaseCompleteReason** со значением **securityDenied** или **OpenLogicalChannelReject** со значением **securityDenied** в **cause**. Отправителю следует выбрать описание защиты, наиболее, по его мнению, безопасное для этих целей.

### 5.5 Упреждающая коррекция ошибок

Для защиты потока FEC, необходимо определить другой главный ключ. Поток FEC и поток медиа-SRTP посылаются на разные IP адреса и/или пару портов, к которым он применяется, как описано в RFC 2733, пункт 11.1. Данный поток FEC должен быть создан с использованием отдельного **OpenLogicalChannel** H.245 с **dataType** – **fec**. Главный ключ для потока FEC должен передаваться в поле **genericKeyMaterial** параметра **secureSharedSecret** (**V3KeySyncMaterial**), содержащегося внутри контейнера **h235Key** в параметре **encryptionSync** сообщения **OpenLogicalChannel** H.245. Главный ключ должен отличаться от всех других главных ключей, предложенных для ассоциируемого медиапотока.

## 6 Шифрование открытым ключом для обеспечения защиты обмена ключами для SRTP

Процедуры шифрования открытым ключом могут быть добавлены для обеспечения сквозной конфиденциальности и аутентификации материала сеансовых ключей SRTP, которыми обмениваются конечные точки H.323, путем зашифровывания и подписывания материала ключа SRTP. Шифрование открытым ключом может быть использовано в случае, где инкапсулирующий протокол защиты, например, IPsec, TLS, завершается на промежуточном устройстве, и поэтому не обеспечивает сквозной защиты.

Сеансовый ключ SRTP, которым зашифровываются медиа-SRTP от вызывающей конечной точки к вызываемой конечной точке, зашифровывается с использованием открытого ключа вызываемой конечной точки и подписывается секретным ключом вызывающей конечной точки. Таким же образом, другой сеансовый ключ SRTP, которым зашифровываются медиа-SRTP от вызываемой конечной точки к вызывающей конечной точке, зашифровывается с использованием открытого ключа вызывающей конечной точки и подписывается секретным ключом вызываемой конечной точки. Процедура, описываемая в данном параграфе, может завершаться на шлюзе или привратнике, а также на конечной точке.

Сеансовый ключ SRTP необходимо передавать с использованием тел Cryptographic Message Syntax (CMS) внутри сообщений H.245. The Cryptographic Message Syntax (RFC 3852) используется для цифровой подписи и шифрования произвольного содержания сообщения. Синтаксис CMS позволяет проводить многочисленные инкапсуляции, при которых один конверт инкапсуляции помещается внутрь другого. В частности, материал сеансового ключа необходимо передавать в теле CMS **EnvelopedData**, под которым ставится подпись с использованием тела CMS **SignedData**.

### 6.1 Идентификация конечной точки

Для идентификации конечной точки, шлюза или привратника в сертификате открытого ключа необходимо использовать следующее:

- H.323 URL;
- URL не стандарта H.323, например *tel*;
- идентификацию/сертификат устройства (FFS).

Сертификат открытого ключа следует использовать для утверждения связи идентичности конечной точки с ее открытым ключом. URL H.323 или URL не стандарта H.323 должен храниться в поле **subjectAltName** сертификата.

Конечные точки могут обслуживать локальное хранилище ключей, в котором содержатся сертификаты открытого ключа других конечных точек, с которыми она (конечная точка) желает установить защищенную сквозную связь. Конечная точка, посылающая подписанное содержание для обеспечения сквозной аутентификации должна включать сертификат ключа, несущий открытый ключ, необходимый для верифицирования подписи. Получающая конечная точка должна либо:

а) верифицировать, что сертификат отправителя подписан утвержденными органами сертификации (CA); либо

б) доверить утверждение защиты сертификату, выданному третьей стороной. Утверждение должно быть подписано универсально верифицируемым материалом ключа.

ПРИМЕЧАНИЕ. – Это может быть полезным в сценариях, где глобальный пользователь PKI не доступен, и используются само, – подписанные сертификаты или сертификаты устройств.

### 6.2 Процедуры обмена ключей SRTP

Если вызывающая и вызываемая конечные точки желают гарантировать сквозную конфиденциальность и аутентификацию материала сеансовых ключей SRTP в случае, когда установление вызова проходит одно или более промежуточных сигнальных устройств, они должны использовать шифрование открытым ключом и обмен сертификатами открытого ключа X.509 (RFC 3280).

Процедуры предложение/ответ, описанные в предыдущих параграфах, остаются без изменений, кроме тех, что описаны ниже.

#### 6.2.1 Обмен возможностями

Для согласования использования сертификатов открытого ключа для обмена ключами SRTP конечная точка H.323 должна установить следующее значение в поле **genericH235SecurityCapability** внутри **encryptionAuthenticationAndIntegrity** в записи **h235SecurityCapability** в **capabilityTable** сообщения **TerminalCapabilitySet** H.245:

- **capabilityIdentifier** должно содержать H.235.8 CMS Object Identifier (см. таблицу 4) в поле **standard**;
- **maxbitRate**, **collapsing**, **nonCollapsing** и **transport** не используются;
- **nonCollapsingRaw** должно содержать параметр **SrtpCryptoCapability**.

#### 6.2.2 Обмен ключами

Если сеансовые ключи SRTP зашифровывается с использованием открытого ключа, зашифрованный сеансовый ключ SRTP передается внутри тел Cryptographic Message Syntax (CMS) в сообщениях H.245. Вместо **SrtpKeys** в поле **genericKeyMaterial** параметра **secureSharedSecret** (**V3KeySyncMaterial**), содержащегося в контейнере **h235Key** в параметре **encryptionSync** сообщений **OpenLogicalChannel** H.245 должны передаваться тела CMS **EnvelopedData** и CMS **SignedData**. Тело CMS **EnvelopedData** необходимо поместить в поле **genericKeyMaterial**, и сразу после него – тело CMS **SignedData**.

Структуру **SrtpKeys** необходимо зашифровать с использованием CMS Content Encryption Key (CEK) и передать в структуре **EncryptedContentInfo** тела CMS **EnvelopedData**.

Присутствие тела CMS, содержащего материал сеансового ключа SRTP в контейнере **genericKeyMaterial**, должно быть обозначено с использованием значения H.235.8 CMS Object Identifier (см. таблицу 4) в поле **standard** поля **capabilityIdentifier** внутри поля **genericH235SecurityCapability** поля **encryptionAuthenticationAndIntegrity** в **h235Media dataType** OLC.

**Таблица 4/H.235.8 – Идентификатор объекта CMS H.235.8**

| **Значение OID** |
|---|
| { itu-t (0) recommendation (0) h (8) 235 version (0) 4 94 } |

### 6.3 Использование тела CMS

Конечная точка, которая генерирует материал сеансового ключа **SrtpKeys** SRTP**,** отправляющая конечная точка, должна зашифровать его, используя ключ шифрования содержимого (CEK) CMS, который сам зашифровывается открытым ключом другой конечной точки, получающей конечной точки. Затем отправляющая конечная точка должна поместить зашифрованный материал сеансового ключа SRTP в тело **EnvelopedData** CMS**.** Затем отправляющая конечная точка должна поставить цифровую подпись под телом **EnvelopedData**, используя свой секретный ключ, и создать "отдельную подпись" тела **SignedData** CMS. Отправляющая конечная точка должна включить сертификат со своим открытым ключом в тело **SignedData** CMS**.** Отправляющая конечная точка должна отправить тело **EnvelopedData** с телом "отдельная подпись" **SignedData** получающей конечной точке. Создание отправляющей конечной точкой тел **EnvelopedData** и **SignedData** более подробно описано в следующих параграфах.

Тела **EnvelopedData** и "отдельная подпись" **SignedData** показаны на рисунке 5.

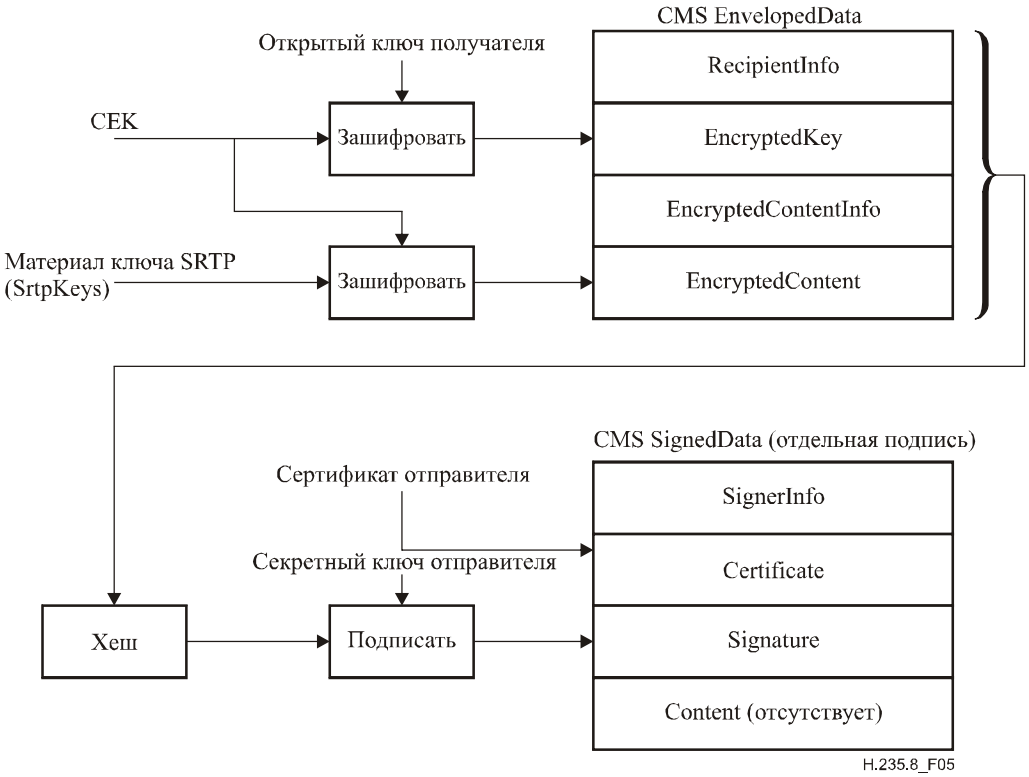


**Рисунок 5/H.235.8 – Тела CMS EnvelopedData и SignedData**

### 6.3.1 Процедуры отсылающей конечной точки

Для генерирования, шифрования и подписи материала сеансового ключа SRTP отправляющая конечная точка должна использовать следующие процедуры.

#### 6.3.1.1 Тело EnvelopedData

Отправляющая конечная точка должна создать тело **EnvelopedData** следующим образом:

1) Сгенерировать материал сеансового ключа SRTP **SrtpKeys** для криптоблока.

2) Сгенерировать случайный ключ ключ шифрования содержимого (CEK).

3) Зашифровать CEK, используя открытый ключ получающей конечной точки. Предполагается, что у отправляющей конечной точки уже есть открытый ключ и сертификат получающей конечной точки. Поместить идентификатор алгоритма, использованного при зашифровывании CEK в поле **keyEncryptionAlgorithm** структуры **RecipientInfo.ktri**.

4) Поместить зашифрованный CEK в поле **encryptedKey** структуры **RecipientInfo** тела **EnvelopedData**. Поле **rid** структуры **RecipientInfo.ktri** используется для идентификации сертификата получающей конечной точки и открытого ключа, который был использован для зашифровывания CEK.

5) Зашифровать материал ключа SRTP **SrtpKeys**, используя CEK, и поместить идентификатор алгоритма, использованного при шифровании в поле **contentEncryptionAlgorithm** структуры **EncryptedContentInfo**.

6) Поместить материал ключа SRTP в поле **encryptedContent** структуры **EncryptedContentInfo**.

#### 6.3.1.2 Тело SignedData

Отправляющая конечная точка должна создать тело "отдельная подпись" **SignedData** следующим образом:

1) Вычислить профиль сообщения или значение хеша по телу **EnvelopedData**. Идентификатор алгоритма профиля сообщения помещается в поле **digestAlgorithm** структуры **SignerInfo**.

2) Поставить цифровую подпись под профилем сообщения, используя секретный ключ отправляющей конечной точки, и поместить значение цифровой подписи в поле **signature** структуры **SignerInfo**. Идентификатор алгоритма цифровой подписи помещается в поле **signatureAlgorithm** структуры **SignerInfo**.

3) Поместить сертификат, содержащий открытый ключ отправляющей конечной точки, в структуру **certificates** структуры **SignerData**. В поле **sid** структуры **SignerInfo** должен отображаться/идентифицироваться сертификат, использующий либо известное имя запрашивающей стороны и серийный номер сертификата, либо значение расширения subjectKeyIdentifier X.509.

4) Поле **eContentType** структуры **encapContentInfo** в теле **SignedData** должно содержать OID id-envelopedData. Поле **eContent** структуры **encapContentInfo** в теле **SignedData** должно отсутствовать, так как это отдельная подпись, а настоящим подписанным содержанием является тело **EnvelopedData**.

### 6.3.2 Процедуры принимающей конечной точки

Для верификации и расшифровки материала сеансового ключа SRTP получающая конечная точка должна выполнить следующие процедуры.

Если при проведении процедур, описанных ниже, получающая конечная точка сталкивается с какими-либо ошибками проверки правильности, вызов отклоняется путем отправки **ReleaseComplete** с **ReleaseCompleteReason** со значением **securityDenied**, или путем отправки элемента **FastConnectRefused** в сообщении H.225.0.

#### 6.3.2.1 Тело SignedData

Получающая конечная точка должна верифицировать полученное тело "detached signature" **SignedData** следующим образом:

1) Извлечь сертификат отправляющей конечной точки из структуры **certificates** структуры **SignerData**.

2) Проверить достоверность сертификата отправляющей конечной точки. Детали проверки правильности пути сертификата не входят в область применения данной Рекомендации. Если получатель не может аутентифицировать отправляющую конечную точку, он может отклонить вызов.

3) Затем получающая конечная точка может добавить проверенный сертификат к своему хранилищу ключей.

4) Верифицировать значение цифровой подписи в поле **signature** структуры **SignerInfo**, используя открытый ключ отправляющей конечной точки из проверенного сертификата. Использовать алгоритм цифровой подписи, описанный в поле **signatureAlgorithm** структуры **SignerInfo**. Результатом дешифровки будет профиль сообщения по телу **EnvelopedData**, вычисленный отправляющей конечной точкой.

5) Вычислить профиль сообщения по полученному телу **EnvelopedData**, используя идентификатор алгоритма профиля сообщения, приведенный в поле **digestAlgorithm** структуры **SignerInfo**.

6) Сравнить значение дешифрованного профиля сообщения с вычисленным значением профиля сообщения. Если значения совпадают, можно начать обработку тела **EnvelopedData**. Если значения не совпадают, получающая конечная точка должна отклонить вызов.

#### 6.3.2.2 Тело EnvelopedData

Получающая конечная точка должна извлекать материал сеансового ключа SRTP из тела **EnvelopedData** следующим образом:

1) Для идентификации сертификата и соответствующего секретного ключа получающей конечной точки в ее хранилище ключей использовать поле **rid** в структуре **RecipientInfo**. Если получающая конечная точка получает тело **EnvelopedData**, зашифрованное неизвестным ей открытым ключом, она должна отклонить вызов.

2) Извлечь зашифрованный CEK из поля **encryptedKey** структуры **RecipientInfo.ktri** тела **EnvelopedData**.

3) Дешифровать зашифрованный CEK, используя секретный ключ получающей конечной точки и алгоритм, описанный в поле **keyEncryptionAlgorithm** структуры **RecipientInfo.ktri**.

4) Извлечь зашифрованный материал сеансового ключа SRTP из зашифрованного материала сеансового ключа SRTP в поле **encryptedContent** структуры **EncryptedContentInfo**.

5) Дешифровать зашифрованный материал сеансового ключа SRTP, используя CEK и алгоритм, описанный в поле **contentEncryptionAlgorithm** структуры **EncryptedContentInfo**.

## 7 Синтаксис описаний защиты SRTP H.235

Синтаксис ASN.1 определяется ниже.

```
H235-SRTP DEFINITIONS AUTOMATIC TAGS::=
BEGIN

IMPORTS
     GenericData
FROM H323-MESSAGES;


 SrtpCryptoCapability::= SEQUENCE OF SrtpCryptoInfo -- used in H.245
 genericH235SecurityCapability


 SrtpCryptoInfo::= SEQUENCE
 {
    cryptoSuite                     OBJECT IDENTIFIER OPTIONAL,
    sessionParams                   SrtpSessionParameters OPTIONAL,
    allowMKI                        BOOLEAN OPTIONAL,
    ...
 }

 SrtpKeys::= SEQUENCE OF SrtpKeyParameters -- used in H.235 V3KeySyncMaterial

 SrtpKeyParameters::= SEQUENCE
 {
    masterKey                       OCTET STRING,
    masterSalt                      OCTET STRING,
    lifetime                        CHOICE
    {
       powerOfTwo                   INTEGER,
       specific                     INTEGER,
       …
    } OPTIONAL,
    mki                             SEQUENCE
    {
       length                       INTEGER(1..128),
       value                        OCTET STRING,

     ...
    } OPTIONAL,
    ...
 }


 SrtpSessionParameters::= SEQUENCE
 {
    kdr                             INTEGER(0..24) OPTIONAL, -- power of 2
    unencryptedSrtp                 BOOLEAN OPTIONAL,
    unencryptedSrtcp                BOOLEAN OPTIONAL,
    unauthenticatedSrtp             BOOLEAN OPTIONAL,
    fecOrder                        FecOrder OPTIONAL,
    windowSizeHint                  INTEGER(64..65535) OPTIONAL,
    newParameter                    SEQUENCE OF GenericData OPTIONAL,
    ...
 }

 FecOrder::= SEQUENCE
 {
    fecBeforeSrtp                   NULL OPTIONAL,
    fecAfterSrtp                    NULL OPTIONAL,
    ...
 }

END
```

## СЕРИИ РЕКОМЕНДАЦИЙ МСЭ-Т

| | |
|---|---|
| Серия A | Организация работы МСЭ-Т |
| Серия D | Общие принципы тарификации |
| Серия E | Общая эксплуатация сети, телефонная служба, функционирование служб и человеческие факторы |
| Серия F | Нетелефонные службы электросвязи |
| Серия G | Системы и среда передачи, цифровые системы и сети |
| **Серия H** | **Аудиовизуальные и мультимедийные системы** |
| Серия I | Цифровая сеть с интеграцией служб |
| Серия J | Кабельные сети и передача сигналов телевизионных и звуковых программ и других мультимедийных сигналов |
| Серия K | Защита от помех |
| Серия L | Конструкция, прокладка и защита кабелей и других элементов линейно-кабельных сооружений |
| Серия M | Управление электросвязью, включая СУЭ и техническое обслуживание сетей |
| Серия N | Техническое обслуживание: международные каналы передачи звуковых и телевизионных программ |
| Серия O | Требования к измерительной аппаратуре |
| Серия P | Качество телефонной передачи, телефонные установки, сети местных линий |
| Серия Q | Коммутация и сигнализация |
| Серия R | Телеграфная передача |
| Серия S | Оконечное оборудование для телеграфных служб |
| Серия T | Оконечное оборудование для телематических служб |
| Серия U | Телеграфная коммутация |
| Серия V | Передача данных по телефонной сети |
| Серия X | Сети передачи данных, взаимосвязь открытых систем и безопасность |
| Серия Y | Глобальная информационная инфраструктура, аспекты межсетевого протокола и сети последующих поколений |
| Серия Z | Языки и общие аспекты программного обеспечения для систем электросвязи |

Отпечатано в Швейцарии
Женева, 2007 г.