Network Working Group
Request for Comments: 4490
Category: Standards Track

S. Leontiev, Ed.
G. Chudov, Ed.
CRYPTO-PRO

May 2006

Использование алгоритмов ГОСТ 28147-89, ГОСТ Р R 34.11-94, ГОСТ Р GOST R 34.10-94 и ГОСТ Р 34.10-2001 с синтаксисом криптографический сообщений (CMS)

Using the GOST 28147-89, GOST R 34.11-94, GOST R 34.10-94, and GOST R 34.10-2001 Algorithms with Cryptographic Message Syntax (CMS)

Статус документа

В этом документе приведена спецификация проекта стандартного протокола Internet. Документ служит приглашением к дискуссии в целях развития и совершенствования протокола. Текущий статус стандартизации протокола можно узнать из текущей версии документа Internet Official Protocol Standards (STD 1). Документ может распространяться без ограничений.

Аннотация

Данный документ описывает соглашения по использованию криптографических алгоритмов GOST 28147-89, GOST R 34.10-94, GOST R 34.10-2001 и GOST R 34.11-94 с синтаксисом криптографических сообщений (CMS[1]). CMS применяется для цифровых подписей, дайджестов, аутентификации и шифрования произвольного содержимого сообщений.

Оглавление

1. Введение

Синтаксис криптографических сообщений [CMS] используется для цифровых подписей, дайджестов, аутентификации и шифрования произвольного содержимого сообщений. Эта дополняющая спецификация описывает использование

[1]Cryptographic Message Syntax.

криптографических алгоритмов GOST 28147-89 [GOST28147], GOST R 34.10-94 [GOST3431095, GOSTR341094], GOST R 34.10-2001 [GOST3431004, GOSTR341001] и GOST R 34.11-94 [GOST3431195, GOSTR341194] в CMS, предложенное компанией «Крипто-Про» в рамках «Соглашения о совместимости СКЗИ». Данный документ не описывает упомянутые криптографические алгоритмы, которые определены в соответствующих национальных стандартах.

Значения CMS создаются с использованием синтаксиса ASN.1 [X.208-88] и представления BER [X.209-88]. Данный документ задаёт идентификаторы для каждого алгоритма, включая ASN.1 для идентификаторов объектов и всех связанных параметров.

Определены поля CMS, используемые каждым алгоритмом.

1.1. Уровни требований

Ключевые слова **необходимо** (MUST), **недопустимо** (MUST NOT), **требуется** (REQUIRED), **нужно** (SHALL), **не нужно** (SHALL NOT), **следует** (SHOULD), **не следует** (SHOULD NOT), **рекомендуется** (RECOMMENDED), **возможно** (MAY), **необязательно** (OPTIONAL) в данном документе интерпретируются в соответствии с [RFC2119].

2. Алгоритмы хеширования сообщений

В этом разделе описаны соглашения для использования алгоритма GOST R 34.11-94 в CMS.

Хэш-значения «отпечатков» (digest) размещаются в поле digest структуры DigestedData и подписанном атрибуте Message Digest. Кроме того, хэш-значения являются входными данными для алгоритмов подписи.

2.1. Алгоритм GOST R 34.11-94

Хэш-функция GOST R 34.11-94 разработана Главным управлением безопасности связи Федерального агентства правительственной связи и информации, а также Всероссийским НИИ Стандартизации. Алгоритм GOST R 34.11-94 даёт на выходе 256-битовое хэш-значение для произвольного конечного размера входных данных. Данный документ не включает полной спецификации GOST R 34.11-94, которую можно найти в [GOSTR341194] на русском языке. В работе [Schneier95], параграф 18.11, стр. 454 приведено краткое техническое описание алгоритма на английском языке.

Идентификатор алгоритма хэширования GOST R 34.11-94 приведён ниже.

```
id-GostR3411-94 OBJECT IDENTIFIER ::=
      { iso(1) member-body(2) ru(643) rans(2) cryptopro(2) gostr3411(9) }
```

В структуре AlgorithmIdentifier **должно** присутствовать поле parameters и это поле **должно** иметь значение NULL. Реализации **могут** воспринимать структуры GOST R 34.11-94 AlgorithmIdentifier как без поля parameters так и со полем, имеющим значение NULL.

Эта функция всегда используется с принятым по умолчанию id-GostR3411-94-CryptoProParamSet (см. параграф 8.2 в [CPALGS]).

При наличии подписанного атрибута хэш DigestedData содержит 32-байтовый «отпечаток» (хэш) в представлении little-endian

```
GostR3411-94-Digest ::= OCTET STRING (SIZE (32))
```

3. Алгоритмы подписи

В этом разделе указаны процедуры CMS для алгоритмов цифровой подписи GOST R 34.10-94 и GOST R 34.10-2001.

Идентификаторы алгоритмов размещаются в поле signatureAlgorithm структуры SignerInfo, вложенной в структуру SignedData. Идентификаторы алгоритма указываются также в поле signatureAlgorithm структуры SignerInfo атрибутов удостоверяющей подписи.

Значения подписи размещаются в поле signature структуры SignerInfo, вложенной в SignedData, а также в поле signature структуры SignerInfo атрибутов удостоверяющей подписи.

3.1. Алгоритм GOST R 34.10-94

Алгоритм GOST R 34.10-94 разработан Главным управлением безопасности связи Федерального агентства правительственной связи и информации, а также Всероссийским НИИ Стандартизации. Этот алгоритм подписи **должен** использоваться совместно с алгоритмом хеширования сообщений GOST R 34.11-94. В данном документе не приводится полной спецификации алгоритма GOST R 34.10-94, которая приведена в [GOSTR341094] на русском языке, а краткое описание на английском языке содержится в работе [Schneier95] (глава 20.3, стр. 495).

Идентификатор алгоритма открытого ключа для алгоритма цифровой подписи GOST R 34.10-94 приведён ниже

```
id-GostR3410-94-signature OBJECT IDENTIFIER ::= id-GostR3410-94
```

Идентификатор id-GostR3410-94 определён в параграфе 2.3.1 [CPPK].

Алгоритм цифровой подписи GOST R 34.10-2001 создаёт подпись в виде двух 256-битовых чисел r' и s. Её представление в форме строки октетов включает 64, из которых первые 32 содержат представление s в формате big-endian, а вторые 32 — представление r' в том же формате.

```
GostR3410-94-Signature ::= OCTET STRING (SIZE (64))
```

3.2. Алгоритм GOST R 34.10-2001

Алгоритм GOST R 34.10-2001 разработан Главным управлением безопасности связи Федерального агентства правительственной связи и информации, а также Всероссийским НИИ Стандартизации. Этот алгоритм **должен** применяться вместе с GOST R 34.11-94. Данный документ не содержит полной спецификации GOST R 34.10-2001, она приведена в [GOSTR341001].

Идентификатор алгоритма открытого ключа для GOST R 34.10-2001 приведён ниже.

```
id-GostR3410-2001-signature OBJECT IDENTIFIER ::= id-GostR3410-2001
```

Определение id-GostR3410-2001 дано в параграфе 2.3.2 документа [CPPK].

Алгоритм цифровой подписи GOST R 34.10-2001 создаёт подпись в виде двух 256-битовых чисел r и s. Её представление в форме строки октетов включает 64 октета, из которых первые 32 содержат представление s в формате big-endian, а вторые 32 — представление r в том же формате.

```
GostR3410-2001-Signature ::= OCTET STRING (SIZE (64))
```

4. Алгоритмы управления ключами

В этом разделе описаны алгоритмы согласования и доставки ключей, основанные на алгоритмах создания производных ключей VKO GOST R 34.10-94 и VKO GOST R 34.10-2001, а также алгоритмах шифрования ключей CryptoPro и GOST 28147-89, описанных в [CPALGS]. Они **должны** применяться только с алгоритмом шифрования содержимого GOST 28147-89, рассмотренным в разделе 5 данного документа.

4.1. Алгоритмы согласования ключей

В этом параграфе указаны соглашения, применяемые реализациями CMS, которые поддерживают согласование ключей с использованием алгоритмов VKO GOST R 34.10-94 и VKO GOST R 34.10-2001, описанных в [CPALGS].

Идентификаторы алгоритма согласования ключей размещаются в полях EnvelopedData RecipientInfos KeyAgreeRecipientInfo keyEncryptionAlgorithm и AuthenticatedData RecipientInfos KeyAgreeRecipientInfo keyEncryptionAlgorithm.

Зашифрованные ключи шифрования содержимого размещаются в поле EnvelopedData RecipientInfos KeyAgreeRecipientInfo RecipientEncryptedKeys encryptedKey. Зашифрованные ключи проверки подлинности сообщений размещаются в поле AuthenticatedData RecipientInfos KeyAgreeRecipientInfo RecipientEncryptedKeys encryptedKey.

4.1.1. Алгоритмы согласования на базе открытых ключей GOST R 34.10-94/2001

Использование поля EnvelopedData RecipientInfos KeyAgreeRecipientInfo описано ниже.

Поле version **должно** иметь значение 3.

В поле originator **должно** помещаться значение originatorKey. Поле algorithm в originatorKey **должно** содержать идентификатор объекта id-GostR3410-94 или id-GostR3410-2001 и соответствующие параметры (определены в параграфах 2.3.1 и 2.3.2 [CPPK]).

В поле originatorKey publicKey **должен** указываться открытый ключ отправителя.

В поле keyEncryptionAlgorithm **должен** помещаться идентификатор алгоритма id-GostR3410-94-CryptoPro-ESDH или id-GostR3410-2001-CryptoPro-ESDH в зависимости от алгоритма открытого ключа получателя. Полем идентификатора параметров для этих алгоритмов является KeyWrapAlgorithm и этот параметр **должен** присутствовать. KeyWrapAlgorithm указывает алгоритм и параметры, применяемые для шифрования ключа шифрования содержимого с помощью парного ключа шифрования ключей, созданного с использованием алгоритма согласования ключей VKO GOST R 34.10-94 или VKO GOST R 34.10-2001.

Синтаксис идентификаторов и параметров алгоритмов показан ниже.

```
id-GostR3410-94-CryptoPro-ESDH OBJECT IDENTIFIER ::=
    { iso(1) member-body(2) ru(643) rans(2) cryptopro(2)
      gostR3410-94-CryptoPro-ESDH(97) }

id-GostR3410-2001-CryptoPro-ESDH OBJECT IDENTIFIER ::=
    { iso(1) member-body(2) ru(643) rans(2) cryptopro(2)
      gostR3410-2001-CryptoPro-ESDH(96) }

KeyWrapAlgorithm ::= AlgorithmIdentifier
```

Если keyEncryptionAlgorithm = id-GostR3410-94-CryptoPro-ESDH, в качестве KeyWrapAlgorithm **должен** указываться идентификатор id-Gost28147-89-CryptoPro-KeyWrap.

```
id-Gost28147-89-CryptoPro-KeyWrap OBJECT IDENTIFIER ::=
    { iso(1) member-body(2) ru(643) rans(2) cryptopro(2) keyWrap(13) cryptoPro(1) }
```

Алгоритм шифрования ключей CryptoPro описан в параграфах 6.3 и 6.4 [CPALGS].

Если keyEncryptionAlgorithm = id-GostR3410-2001-CryptoPro-ESDH, в качестве KeyWrapAlgorithm **должен** указываться идентификатор алгоритма id-Gost28147-89-CryptoPro-KeyWrap или id-Gost28147-89-None-KeyWrap.

```
id-Gost28147-89-None-KeyWrap OBJECT IDENTIFIER ::=
    { iso(1) member-body(2) ru(643) rans(2) cryptopro(2) keyWrap(13) none(0) }
```

Алгоритм шифрования ключей ГОСТ 28147-89 описан в параграфах 6.1 и 6.2 [CPALGS].

Должны присутствовать параметры алгоритма KeyWrapAlgorithm, синтаксис которых показан ниже.

```
Gost28147-89-KeyWrapParameters ::=
  SEQUENCE {
      encryptionParamSet Gost28147-89-ParamSet,
      ukm                OCTET STRING (SIZE (8)) OPTIONAL
  }
  Gost28147-89-ParamSet ::= OBJECT IDENTIFIER
```

Поле ukm в структуре Gost28147-89-KeyWrapParameters **должно** отсутствовать.

В структуре KeyAgreeRecipientInfo поле ukm **должно** присутствовать и включать 8 октетов.

Поле encryptedKey **должно** инкапсулировать Gost28147-89-EncryptedKey, где поле maskKey **должно** отсутствовать.

```
Gost28147-89-EncryptedKey ::=   SEQUENCE {
```

```
        encryptedKey          Gost28147-89-Key,
        maskKey               [0] IMPLICIT Gost28147-89-Key OPTIONAL,
        macKey                Gost28147-89-MAC
      }
```

Для получения ключа шифрования KEK используется секретный ключ, соответствующий originatorKey publicKey, и открытый ключу получателя с алгоритмом VKO GOST R 34.10-94 или VKO GOST R 34.10-2001 (описаны в [CPALGS]).

Затем применяется алгоритм шифрования ключа, указанный в KeyWrapAlgorithm, для получения CEK_ENC, CEK_MAC, и UKM. Для всех операций шифрования применяется набор параметров encryptionParamSet структуры Gost28147-89-KeyWrapParameters.

Полученный в результате зашифрованный ключ (CEK_ENC) помещается в поле encryptedKey структуры Gost28147-89-EncryptedKey, код mac (CEK_MAC) в поле macKey этой структуры, а UKM в поле ukm структуры KeyAgreeRecipientInfo.

4.2. Алгоритмы доставки ключей

В этом разделе описаны соглашения, используемые реализациями CMS, которые поддерживают доставку ключей с помощью алгоритмов VKO GOST R 34.10-94 и VKO GOST R 34.10-2001, описанных [CPALGS].

Идентификатор алгоритма доставки ключей помещается в поле EnvelopedData RecipientInfos KeyTransRecipientInfo keyEncryptionAlgorithm.

Зашифрованный ключ шифрования содержимого для транспортировки помещается в поле EnvelopedData RecipientInfos KeyTransRecipientInfo encryptedKey.

4.2.1. Доставка на базе открытых ключей GOST R 34.10-94/2001

Использование поля EnvelopedData RecipientInfos KeyTransRecipientInfo рассмотрено ниже.

Поле version **должно** иметь значение 0 или 3.

keyEncryptionAlgorithm и параметры алгоритма **должны** совпадать с алгоритмом открытого ключа получателя и параметрами этого алгоритма.

encryptedKey включает в себя структуру GostR3410-KeyTransport, состоящую из зашифрованного ключа шифрования содержимого, значения MAC для него, параметров алгоритма GOST 28147-89, использованных при шифровании ключа, эфемерного открытого ключа отправителя и ключевого материала UKM (UserKeyingMaterial, см параграф 10.2.6 [CMS]).

Поле transportParameters **должно** присутствовать.

Поле ephemeralPublicKey **должно** присутствовать, а его параметры (при наличии) **должны** совпадать с параметрами открытого ключа получателя.

```
   GostR3410-KeyTransport ::= SEQUENCE {
     sessionEncryptedKey   Gost28147-89-EncryptedKey,
     transportParameters
       [0] IMPLICIT GostR3410-TransportParameters OPTIONAL
   }

   GostR3410-TransportParameters ::= SEQUENCE {
     encryptionParamSet   OBJECT IDENTIFIER,
     ephemeralPublicKey   [0] IMPLICIT SubjectPublicKeyInfo OPTIONAL,
     ukm                  OCTET STRING
   }
```

Ключ шифрования ключей KEK получается с использованием секретного ключа, соответствующего GostR3410-TransportParameters ephemeralPublicKey и открытому ключу получателя, на основе алгоритма VKO GOST R 34.10-94 или VKO GOST R 34.10-2001 (см. [CPALGS]).

Затем используется алгоритм CryptoPro для создания CEK_ENC, CEK_MAC и UKM с параметрами GostR3410-TransportParameters encryptionParamSet для всех операций шифрования.

Полученный в результате зашифрованный ключ (CEK_ENC) помещается в поле Gost28147-89-EncryptedKey encryptedKey, значение mac для него (CEK_MAC) — в поле Gost28147-89-EncryptedKey macKey, а UKM - в поле GostR3410-TransportParameters ukm.

5. Алгоритм шифрования содержимого

В этом разделе описаны соглашения, используемые реализациями CMS с поддержкой шифрования содержимого на основе алгоритма GOST 28147-89.

Идентификаторы алгоритма шифрования содержимого помещаются в поля EnvelopedData EncryptedContentInfo contentEncryptionAlgorithm и EncryptedData EncryptedContentInfo contentEncryptionAlgorithm.

Алгоритмы шифрования содержимого применяются для шифрования данных в полях EnvelopedData EncryptedContentInfo encryptedContent и EncryptedData EncryptedContentInfo encryptedContent.

5.1. Алгоритм GOST 28147-89

В этом параграфе описано использование алгоритма GOST 28147-89 для шифрования данных.

Алгоритм GOST 28147-89 полностью описан в документе [GOST28147] (на русском языке).

Этот документ задаёт для алгоритма идентификатор объекта (OID)

```
id-Gost28147-89 OBJECT IDENTIFIER ::=
      { iso(1) member-body(2) ru(643) rans(2) cryptopro(2) gost28147-89(21) }
```

Параметры алгоритма **должны** присутствовать и имеют приведённую ниже структуру.

```
Gost28147-89-Parameters ::=
  SEQUENCE {
    iv                  Gost28147-89-IV,
    encryptionParamSet  OBJECT IDENTIFIER
   }

Gost28147-89-IV ::= OCTET STRING (SIZE (8))
```

Набор encryptionParamSet задаёт соответствующие параметры Gost28147-89-ParamSetParameters (см. параграф 8.1 в [CPALGS]).

6. Алгоритмы MAC

В этом разделе описаны соглашения, используемые реализациями CMS, которые поддерживают коды аутентификации сообщений (MAC[1]) на основе GOST R 34.11-94.

Идентификатор алгоритма MAC указывается в поле macAlgorithm структуры AuthenticatedData.

Значения кодов MAC помещаются в поле mac структуры AuthenticatedData.

6.1. HMAC с GOST R 34.11-94

Функция HMAC_GOSTR3411(K,text) основана на хэш-функции GOST R 34.11-94, как определено в разделе 3 [CPALGS].

Идентификатор объекта (OID) для этого алгоритма приведён ниже.

```
id-HMACGostR3411-94 OBJECT IDENTIFIER ::=
      { iso(1) member-body(2) ru(643) rans(2) cryptopro(2) hmacgostr3411(10) }
```

Данный алгоритм использует такие же параметры как алгоритм хеширования GOST R 34.11-94 и те же значения OID для их идентификации (см. [CPPK]).

7. Использование с S/MIME

В этом разделе описано применение определенных в данном документе алгоритмов совместно с S/MIME [RFC3851].

7.1. Параметр micalg

При использовании определенных в этом документе алгоритмов для параметра micalg **следует** устанавливать значение gostr3411-94, в остальных случаях **должно** указываться значение unknown.

7.2. Атрибут SMIMECapabilities

Значение SMIMECapability, указывающее поддержку алгоритма хэширования GOST R 34.11-94, представляет собой последовательность (SEQUENCE) с полем capabilityID, содержащим идентификатор алгоритма id-GostR3411-94 без параметров. DER-представление имеет вид

```
30 08 06 06  2A 85 03 02  02 09
```

Значение SMIMECapability, указывающее поддержку алгоритма шифрования GOST 28147-89, представляет собой последовательность (SEQUENCE) с полем capabilityID, содержащим идентификатор алгоритма id-Gost28147-89 без параметров. DER-представление имеет вид

```
30 08 06 06  2A 85 03 02  02 15
```

Если отправитель желает указать поддержку конкретного набора параметров, параметры SMIMECapability **должны** содержать структуру Gost28147-89-Parameters. Получатели **должны** игнорировать поле iv в структуре Gost28147-89-Parameters и предполагать, что отправитель поддерживает параметры, указанные в поле encryptionParamSet структуры Gost28147-89-Parameters.

DER-представление для SMIMECapability с индикацией поддержки GOST 28147-89 с id-Gost28147-89-CryptoPro-A-ParamSet (см. [CPALGS]) имеет вид

```
30 1D 06 06  2A 85 03 02  02 15 30 13  04 08 00 00
00 00 00 00  00 00 06 07  2A 85 03 02  02 1F 01
```

8. Вопросы безопасности

Соответствующие данной спецификации приложения **должны** использовать уникальные значения для полей ukm и iv. Получатели **могут** проверять уникальность заданных отправителем значений ukm и iv.

Программным приложениям **рекомендуется** проверять соответствие значений подписи, открытых ключей субъекта и параметров алгоритма стандартам [GOSTR341001] и [GOSTR341094] до их использования.

Параметры криптографического алгоритма влияют на его стойкость. Использовать параметры, не указанные в [CPALGS] **не рекомендуется** (см. раздел «Вопросы безопасности» в [CPALGS]).

Использовать один и тот же ключ для подписи и создания ключей **не рекомендуется**. При использовании подписанных документов CMS в качестве аналога подписанных человеком документов в контексте российского законодательства об электронной подписи [RFEDSL] сертификат подписывающей стороны **должен** включать расширение keyUsage, которое **должно** быть критическим, а включение keyEncipherment или keyAgreement в keyUsage **недопустимо** (см. [PROFILE], параграф 4.2.1.3). Приложение **следует** подавать на экспертизу в уполномоченную организацию для проверки в соответствии с целью использования и требованиями [RFEDSL], [RFLLIC] и [CRYPTOLIC].

[1] Message authentication code.

9. Примеры

В приведённых здесь примерах используется тот же формат записи, что и в примерах [RFC4134] и для извлечения кода может применяться та же программа.

Если вы хотите извлечь код без использования программы, скопируйте все строки между маркерами |> и |<, удалите символы перевода страниц, а также символы | в начале каждой строки. В результате будет получен корректный блок данных Base64, который можно обработать любым декодером Base64.

9.1. Подписанное сообщение

Это сообщение подписано с использованием примера сертификата из параграфа 4.2 [CPPK]. Для проверки подписи сообщения можно использовать открытый ключ key (x,y) из того же параграфа.

```
   0  296: SEQUENCE {
   4    9:  OBJECT IDENTIFIER signedData
  15  281:  [0] {
  19  277:   SEQUENCE {
  23    1:    INTEGER 1
  26   12:    SET {
  28   10:     SEQUENCE {
  30    6:      OBJECT IDENTIFIER id-GostR3411-94
  38    0:      NULL
         :      }
         :     }
  40   27:    SEQUENCE {
  42    9:     OBJECT IDENTIFIER data
  53   14:     [0] {
  55   12:      OCTET STRING 73 61 6D 70 6C 65 20 74 65 78 74 0A
         :      }
         :     }
  69  228:    SET {
  72  225:     SEQUENCE {
  75    1:      INTEGER 1
  78  129:      SEQUENCE {
  81  109:       SEQUENCE {
  83   31:        SET {
  85   29:         SEQUENCE {
  87    3:          OBJECT IDENTIFIER commonName
  92   22:          UTF8String 'GostR3410-2001 example'
         :          }
         :         }
 116   18:        SET {
 118   16:         SEQUENCE {
 120    3:          OBJECT IDENTIFIER organizationName
 125    9:          UTF8String 'CryptoPro'
         :          }
         :         }
 136   11:        SET {
 138    9:         SEQUENCE {
 140    3:          OBJECT IDENTIFIER countryName
 145    2:          PrintableString 'RU'
         :          }
         :         }
 149   41:        SET {
 151   39:         SEQUENCE {
 153    9:          OBJECT IDENTIFIER emailAddress
 164   26:          IA5String 'GostR3410-2001@example.com'
         :          }
         :         }
         :        }
 192   16:       INTEGER
         :        2B F5 C6 1E C2 11 BD 17 C7 DC D4 62 66 B4 2E 21
         :       }
 210   10:      SEQUENCE {
 212    6:       OBJECT IDENTIFIER id-GostR3411-94
 220    0:       NULL
         :       }
 222   10:      SEQUENCE {
 224    6:       OBJECT IDENTIFIER id-GostR3410-2001
 232    0:       NULL
         :       }
 234   64:      OCTET STRING
         :       C0 C3 42 D9 3F 8F FE 25 11 11 88 77 BF 89 C3 DB
         :       83 42 04 D6 20 F9 68 2A 99 F6 FE 30 3B E4 F4 C8
         :       F8 D5 B4 DA FB E1 C6 91 67 34 1F BC A6 7A 0D 12
         :       7B FD 10 25 C6 51 DB 8D B2 F4 8C 71 7E ED 72 A9
         :      }
         :     }
         :    }
         :   }
         :  }

|>GostR3410-2001-signed.bin
|MIIBKAYJKoZIhvcNAQcCoIIBGTCCARUCAQExDDAKBgYqhQMCAgkFADAbBgkqhkiG
|9w0BBwGgDgQMc2FtcGxlIHRleHQKMYHkMIHhAgEBMIGBMG0xHzAdBgNVBAMMFkdv
```

```
|c3RSMzQxMC0yMDAxIGV4YW1wbGUxEjAQBgNVBAoMCUNyeXB0b1BybzELMAkGA1UE
|BhMCUlUxKTAnBgkqhkiG9w0BCQEWGkdvc3RSMzQxMC0yMDAxQGV4YW1wbGUuY29t
|AhAr9cYewhG9F8fc1GJmtC4hMAoGBiqFAwICCQUAMAoGBiqFAwICEwUABEDAw0LZ
|P4/+JRERiHe/icPbg0IE1iD5aCqZ9v4wO+T0yPjVtNr74caRZzQfvKZ6DRJ7/RAl
|xlHbjbL0jHF+7XKp
|<GostR3410-2001-signed.bin
```

9.2. Упакованное с согласованием ключей сообщение

Это сообщение зашифровано с использованием примера сертификата из параграфа 4.2 [CPPK] в качестве сертификата получателя. Для расшифровки сообщения можно использовать секретный ключ d из того же параграфа.

```
  0  420: SEQUENCE {
  4    9:  OBJECT IDENTIFIER envelopedData
 15  405:  [0] {
 19  401:   SEQUENCE {
 23    1:    INTEGER 2
 26  336:    SET {
 30  332:     [1] {
 34    1:      INTEGER 3
 37  101:      [0] {
 39   99:       [1] {
 41   28:        SEQUENCE {
 43    6:         OBJECT IDENTIFIER id-GostR3410-2001
 51   18:         SEQUENCE {
 53    7:          OBJECT IDENTIFIER
       :           id-GostR3410-2001-CryptoPro-XchA-ParamSet
 62    7:          OBJECT IDENTIFIER
       :           id-GostR3411-94-CryptoProParamSet
       :          }
       :         }
 71   67:        BIT STRING, encapsulates {
 74   64:         OCTET STRING
       :          B3 55 39 F4 67 81 97 2B A5 C4 D9 84 1F 27 FB 81
       :          ED 08 32 E6 9A D4 F2 00 78 B8 FF 83 64 EA D2 1D
       :          B0 78 3C 7D FE 03 C1 F4 06 E4 3B CC 16 B9 C5 F6
       :          F6 19 37 1C 17 B8 A0 AA C7 D1 A1 94 B3 A5 36 20
       :         }
       :        }
       :       }
140   10:      [1] {
142    8:       OCTET STRING 2F F0 F6 D1 86 4B 32 8A
       :       }
152   30:      SEQUENCE {
154    6:       OBJECT IDENTIFIER id-GostR3410-2001-CryptoPro-ESDH
162   20:       SEQUENCE {
164    7:        OBJECT IDENTIFIER id-Gost28147-89-None-KeyWrap
173    9:        SEQUENCE {
175    7:         OBJECT IDENTIFIER
       :          id-Gost28147-89-CryptoPro-A-ParamSet
       :         }
       :        }
       :       }
184  179:      SEQUENCE {
187  176:       SEQUENCE {
190  129:        SEQUENCE {
193  109:         SEQUENCE {
195   31:          SET {
197   29:           SEQUENCE {
199    3:            OBJECT IDENTIFIER commonName
204   22:            UTF8String 'GostR3410-2001 example'
       :            }
       :           }
228   18:          SET {
230   16:           SEQUENCE {
232    3:            OBJECT IDENTIFIER organizationName
237    9:            UTF8String 'CryptoPro'
       :            }
       :           }
248   11:          SET {
250    9:           SEQUENCE {
252    3:            OBJECT IDENTIFIER countryName
257    2:            PrintableString 'RU'
       :            }
       :           }
261   41:          SET {
263   39:           SEQUENCE {
265    9:            OBJECT IDENTIFIER emailAddress
276   26:            IA5String 'GostR3410-2001@example.com'
       :            }
       :           }
       :          }
304   16:         INTEGER
       :          2B F5 C6 1E C2 11 BD 17 C7 DC D4 62 66 B4 2E 21
       :         }
322   42:        OCTET STRING, encapsulates {
324   40:         SEQUENCE {
```

```
 326   32:           OCTET STRING
         :            16 A3 1C E7 CE 4E E9 0D F1 EC 74 69 04 68 1E C7
         :            9F 3A ED B8 3B 1F 1D 4A 7E F9 A5 D9 CB 19 D5 E8
 360    4:           OCTET STRING
         :            93 FD 86 7E
         :           }
         :          }
         :         }
         :        }
         :       }
         :      }
 366   56:     SEQUENCE {
 368    9:      OBJECT IDENTIFIER data
 379   29:      SEQUENCE {
 381    6:       OBJECT IDENTIFIER id-Gost28147-89
 389   19:       SEQUENCE {
 391    8:        OCTET STRING B7 35 E1 7A 07 35 A2 1D
 401    7:        OBJECT IDENTIFIER id-Gost28147-89-CryptoPro-A-ParamSet
         :        }
         :       }
 410   12:      [0] 39 B1 8A F4 BF A9 E2 65 25 B6 55 C9
         :      }
         :     }
         :    }
         :   }
```

```
|>GostR3410-2001-keyagree.bin
|MIIBpAYJKoZIhvcNAQcDoIIBlTCCAZECAQIxggFQoYIBTAIBA6BloWMwHAYGKoUD
|AgITMBIGByqFAwICJAAGByqFAwICHgEDQwAEQLNVOfRngZcrpcTZhB8n+4HtCDLm
|mtTyAHi4/4Nk6tIdsHg8ff4DwfQG5DvMFrnF9vYZNxwXuKCqx9GhlLOlNiChCgQI
|L/D20YZLMoowHgYGKoUDAgJgMBQGByqFAwICDQAwCQYHKoUDAgIfATCBszCBsDCB
|gTBtMR8wHQYDVQQDDBZHb3N0UjM0MTAtMjAwMSBleGFtcGxlMRIwEAYDVQQKDAlD
|cnlwdG9Qcm8xCzAJBgNVBAYTAlJVMSkwJwYJKoZIhvcNAQkBFhpHb3N0UjM0MTAt
|MjAwMUBleGFtcGxlLmNvbQIQK/XGHsIRvRfH3NRiZrQuIQQqMCgEIBajHOfOTukN
|8ex0aQRoHsefOu24Ox8dSn75pdnLGdXoBAST/YZ+MDgGCSqGSIb3DQEHATAdBgYq
|hQMCAhUwEwQItzXhegc1oh0GByqFAwICHwGADDmxivS/qeJlJbZVyQ==
|<GostR3410-2001-keyagree.bin
```

9.3. Упакованное с доставкой ключей сообщение

Это сообщение зашифровано с использованием примера сертификата из параграфа 4.2 [CPPK] в качестве сертификата получателя. Для расшифровки сообщения можно использовать секретный ключ d из того же параграфа.

```
   0  423: SEQUENCE {
   4    9:  OBJECT IDENTIFIER envelopedData
  15  408:  [0] {
  19  404:   SEQUENCE {
  23    1:    INTEGER 0
  26  339:    SET {
  30  335:     SEQUENCE {
  34    1:      INTEGER 0
  37  129:      SEQUENCE {
  40  109:       SEQUENCE {
  42   31:        SET {
  44   29:         SEQUENCE {
  46    3:          OBJECT IDENTIFIER commonName
  51   22:          UTF8String 'GostR3410-2001 example'
         :          }
         :         }
  75   18:        SET {
  77   16:         SEQUENCE {
  79    3:          OBJECT IDENTIFIER organizationName
  84    9:          UTF8String 'CryptoPro'
         :          }
         :         }
  95   11:        SET {
  97    9:         SEQUENCE {
  99    3:          OBJECT IDENTIFIER countryName
 104    2:          PrintableString 'RU'
         :          }
         :         }
 108   41:        SET {
 110   39:         SEQUENCE {
 112    9:          OBJECT IDENTIFIER emailAddress
 123   26:          IA5String 'GostR3410-2001@example.com'
         :          }
         :         }
         :        }
 151   16:       INTEGER
         :        2B F5 C6 1E C2 11 BD 17 C7 DC D4 62 66 B4 2E 21
         :       }
 169   28:      SEQUENCE {
 171    6:       OBJECT IDENTIFIER id-GostR3410-2001
 179   18:       SEQUENCE {
 181    7:        OBJECT IDENTIFIER
         :         id-GostR3410-2001-CryptoPro-XchA-ParamSet
 190    7:        OBJECT IDENTIFIER
```

```
        :          id-GostR3411-94-CryptoProParamSet
        :         }
        :        }
 199  167:      OCTET STRING, encapsulates {
 202  164:       SEQUENCE {
 205   40:        SEQUENCE {
 207   32:         OCTET STRING
        :          6A 2F A8 21 06 95 68 9F 9F E4 47 AA 9E CB 61 15
        :          2B 7E 41 60 BC 5D 8D FB F5 3D 28 1B 18 9A F9 75
 241    4:         OCTET STRING
        :          36 6D 98 B7
        :         }
 247  120:        [0] {
 249    7:         OBJECT IDENTIFIER
        :          id-Gost28147-89-CryptoPro-A-ParamSet
 258   99:         [0] {
 260   28:          SEQUENCE {
 262    6:           OBJECT IDENTIFIER id-GostR3410-2001
 270   18:           SEQUENCE {
 272    7:            OBJECT IDENTIFIER
        :             id-GostR3410-2001-CryptoPro-XchA-ParamSet
 281    7:            OBJECT IDENTIFIER
        :             id-GostR3411-94-CryptoProParamSet
        :            }
        :           }
 290   67:          BIT STRING encapsulates {
 293   64:           OCTET STRING
        :            4D 2B 2F 33 90 E6 DC A3 DD 55 2A CD DF E0 EF FB
        :            31 F7 73 7E 4E FF BF 78 89 8A 2B C3 CD 31 94 04
        :            4B 0E 60 48 96 1F DB C7 5D 12 6F DA B2 40 8A 77
        :            B5 BD EA F2 EC 34 CB 23 9F 9B 8B DD 9E 12 C0 F6
        :           }
        :          }
 359    8:         OCTET STRING
        :          97 95 E3 2C 2B AD 2B 0C
        :         }
        :        }
        :       }
        :      }
        :     }
 369   56:    SEQUENCE {
 371    9:     OBJECT IDENTIFIER data
 382   29:     SEQUENCE {
 384    6:      OBJECT IDENTIFIER id-Gost28147-89
 392   19:      SEQUENCE {
 394    8:       OCTET STRING BC 10 8B 1F 0B FF 34 29
 404    7:       OBJECT IDENTIFIER id-Gost28147-89-CryptoPro-A-ParamSet
        :       }
        :      }
 413   12:     [0] AA 8E 72 1D EE 4F B3 2E E3 0F A1 37
        :     }
        :    }
        :   }
        :  }
```

```
|>GostR3410-2001-keytrans.bin
|MIIBpwYJKoZIhvcNAQcDoIIBmDCCAZQCAQAxggFTMIIBTwIBADCBgTBtMR8wHQYD
|VQQDDBZHb3N0UjM0MTAtMjAwMSBleGFtcGxlMRIwEAYDVQQKDAlDcnlwdG9Qcm8x
|CzAJBgNVBAYTAlJVMSkwJwYJKoZIhvcNAQkBFhpHb3N0UjM0MTAtMjAwMUBleGFt
|cGxlLmNvbQIQK/XGHsIRvRfH3NRiZrQuITAcBgYqhQMCAhMwEgYHKoUDAgIkAAYH
|KoUDAgIeAQSBpzCBpDAoBCBqL6ghBpVon5/kR6qey2EVK35BYLxdjfv1PSgbGJr5
|dQQENm2Yt6B4BgcqhQMCAh8BoGMwHAYGKoUDAgITMBIGByqFAwICJAAGByqFAwIC
|HgEDQwAEQE0rLzOQ5tyj3VUqzd/g7/sx93N+Tv+/eImKK8PNMZQESw5gSJYf28dd
|Em/askCKd7W96vLsNMsjn5uL3Z4SwPYECJeV4ywrrSsMMDgGCSqGSIb3DQEHATAd
|BgYqhQMCAhUwEwQIvBCLHwv/NCkGByqFAwICHwGADKqOch3uT7Mu4w+hNw==
|<GostR3410-2001-keytrans.bin
```

10. Модули ASN.1

Дополнительные модули ASN.1, упомянутые здесь, можно найти в [CPALGS].

10.1. GostR3410-EncryptionSyntax

```
GostR3410-EncryptionSyntax
    { iso(1) member-body(2) ru(643) rans(2) cryptopro(2)
      other(1) modules(1) gostR3410-EncryptionSyntax(5) 2 }
DEFINITIONS ::=
BEGIN
-- EXPORTS All --
-- The types and values defined in this module are exported for
-- use in the other ASN.1 modules contained within the Russian
-- Cryptography "GOST" & "GOST R" Specifications, and for the use
-- of other applications which will use them to access Russian
-- Cryptography services.  Other applications may use them for
-- their own purposes, but this will not constrain extensions and
-- modifications needed to maintain or improve the Russian
```

```
-- Cryptography service.
    IMPORTS
        id-CryptoPro-algorithms,
        gost28147-89-EncryptionSyntax,
        gostR3410-94-PKISyntax,
        gostR3410-2001-PKISyntax,
        ALGORITHM-IDENTIFIER,
        cryptographic-Gost-Useful-Definitions
        FROM Cryptographic-Gost-Useful-Definitions -- in [CPALGS]
            { iso(1) member-body(2) ru(643) rans(2)
              cryptopro(2) other(1) modules(1)
              cryptographic-Gost-Useful-Definitions(0) 1 }
        id-GostR3410-94
        FROM GostR3410-94-PKISyntax -- in [CPALGS]
            gostR3410-94-PKISyntax
        id-GostR3410-2001
        FROM GostR3410-2001-PKISyntax -- in [CPALGS]
            gostR3410-2001-PKISyntax
        Gost28147-89-ParamSet,
        Gost28147-89-EncryptedKey
        FROM Gost28147-89-EncryptionSyntax -- in [CPALGS]
             gost28147-89-EncryptionSyntax
        SubjectPublicKeyInfo
        FROM PKIX1Explicit88 {iso(1) identified-organization(3)
        dod(6) internet(1) security(5) mechanisms(5) pkix(7)
        id-mod(0) id-pkix1-explicit-88(1)}
    ;
  -- CMS/PKCS#7 key agreement algorithms & parameters
    Gost28147-89-KeyWrapParameters ::=
      SEQUENCE {
        encryptionParamSet Gost28147-89-ParamSet,
        ukm                OCTET STRING (SIZE (8)) OPTIONAL
      }
    id-Gost28147-89-CryptoPro-KeyWrap OBJECT IDENTIFIER ::=
      { id-CryptoPro-algorithms keyWrap(13) cryptoPro(1) }
    id-Gost28147-89-None-KeyWrap OBJECT IDENTIFIER ::=
      { id-CryptoPro-algorithms keyWrap(13) none(0) }
    Gost28147-89-KeyWrapAlgorithms  ALGORITHM-IDENTIFIER ::= {
      { Gost28147-89-KeyWrapParameters IDENTIFIED BY
        id-Gost28147-89-CryptoPro-KeyWrap } |
      { Gost28147-89-KeyWrapParameters IDENTIFIED BY
        id-Gost28147-89-None-KeyWrap }
    }
    id-GostR3410-2001-CryptoPro-ESDH OBJECT IDENTIFIER ::=
      { id-CryptoPro-algorithms
        gostR3410-2001-CryptoPro-ESDH(96) }
    id-GostR3410-94-CryptoPro-ESDH OBJECT IDENTIFIER ::=
      { id-CryptoPro-algorithms
        gostR3410-94-CryptoPro-ESDH(97) }
  -- CMS/PKCS#7 key transport algorithms & parameters
    -- OID for CMS/PKCS#7 Key transport is id-GostR3410-94 from
    --      GostR3410-94-PKISyntax or id-GostR3410-2001 from
    --      GostR3410-2001-PKISyntax
    -- Algorithms for CMS/PKCS#7 Key transport are
    --      GostR3410-94-PublicKeyAlgorithms from
    --      GostR3410-94-PKISyntax or
    --      GostR3410-2001-PublicKeyAlgorithms from
    --      GostR3410-2001-PKISyntax
    -- SMIMECapability for CMS/PKCS#7 Key transport are
    --      id-GostR3410-94 from GostR3410-94-PKISyntax or
    --      id-GostR3410-2001 from GostR3410-2001-PKISyntax
    id-GostR3410-94-KeyTransportSMIMECapability
        OBJECT IDENTIFIER ::= id-GostR3410-94
    id-GostR3410-2001-KeyTransportSMIMECapability
        OBJECT IDENTIFIER ::= id-GostR3410-2001
    GostR3410-KeyTransport ::=
        SEQUENCE {
            sessionEncryptedKey Gost28147-89-EncryptedKey,
            transportParameters [0]
                IMPLICIT GostR3410-TransportParameters OPTIONAL
        }
    GostR3410-TransportParameters ::=
        SEQUENCE {
            encryptionParamSet Gost28147-89-ParamSet,
            ephemeralPublicKey [0]
                IMPLICIT SubjectPublicKeyInfo OPTIONAL,
            ukm                OCTET STRING ( SIZE(8) )
        }
END -- GostR3410-EncryptionSyntax
```

10.2. GostR3410-94-SignatureSyntax

```
GostR3410-94-SignatureSyntax
    { iso(1) member-body(2) ru(643) rans(2) cryptopro(2)
      other(1) modules(1) gostR3410-94-SignatureSyntax(3) 1 }
DEFINITIONS ::=
```

```
BEGIN
-- EXPORTS All --
-- The types and values defined in this module are exported for
-- use in the other ASN.1 modules contained within the Russian
-- Cryptography "GOST" & "GOST R" Specifications, and for the use
-- of other applications which will use them to access Russian
-- Cryptography services.  Other applications may use them for
-- their own purposes, but this will not constrain extensions and
-- modifications needed to maintain or improve the Russian
-- Cryptography service.
    IMPORTS
        gostR3410-94-PKISyntax, ALGORITHM-IDENTIFIER,
        cryptographic-Gost-Useful-Definitions
        FROM Cryptographic-Gost-Useful-Definitions -- in [CPALGS]
            { iso(1) member-body(2) ru(643) rans(2)
              cryptopro(2) other(1) modules(1)
              cryptographic-Gost-Useful-Definitions(0) 1 }
        id-GostR3410-94,
        GostR3410-94-PublicKeyParameters
        FROM GostR3410-94-PKISyntax -- in [CPALGS]
            gostR3410-94-PKISyntax
    ;
  -- GOST R 34.10-94 signature data type
    GostR3410-94-Signature ::=
        OCTET STRING (SIZE (64))
  -- GOST R 34.10-94 signature algorithm & parameters
    GostR3410-94-CMSSignatureAlgorithms  ALGORITHM-IDENTIFIER ::= {
        { GostR3410-94-PublicKeyParameters IDENTIFIED BY
                        id-GostR3410-94 }
    }
END -- GostR3410-94-SignatureSyntax
```

10.3. GostR3410-2001-SignatureSyntax

```
GostR3410-2001-SignatureSyntax
    { iso(1) member-body(2) ru(643) rans(2) cryptopro(2)
      other(1) modules(1) gostR3410-2001-SignatureSyntax(10) 1 }
DEFINITIONS ::=
BEGIN
-- EXPORTS All --
-- The types and values defined in this module are exported for
-- use in the other ASN.1 modules contained within the Russian
-- Cryptography "GOST" & "GOST R" Specifications, and for the use
-- of other applications which will use them to access Russian
-- Cryptography services. Other applications may use them for
-- their own purposes, but this will not constrain extensions and
-- modifications needed to maintain or improve the Russian
-- Cryptography service.
    IMPORTS
        gostR3410-2001-PKISyntax, ALGORITHM-IDENTIFIER,
        cryptographic-Gost-Useful-Definitions
        FROM Cryptographic-Gost-Useful-Definitions -- in [CPALGS]
            { iso(1) member-body(2) ru(643) rans(2)
              cryptopro(2) other(1) modules(1)
              cryptographic-Gost-Useful-Definitions(0) 1 }
        id-GostR3410-2001,
        GostR3410-2001-PublicKeyParameters -- in [CPALGS]
        FROM GostR3410-2001-PKISyntax
            gostR3410-2001-PKISyntax
    ;
  -- GOST R 34.10-2001 signature data type
    GostR3410-2001-Signature ::=
        OCTET STRING (SIZE (64))
  -- GOST R 34.10-2001 signature algorithms and parameters
    GostR3410-2001-CMSSignatureAlgorithms

        ALGORITHM-IDENTIFIER ::= {
                { GostR3410-2001-PublicKeyParameters IDENTIFIED BY
                        id-GostR3410-2001 }
        }
END -- GostR3410-2001-SignatureSyntax
```

11. Благодарности

Этот документ был подготовлен в соответствии с «Соглашением о совместимости СКЗИ», подписанным ФГУП НТЦ «Атлас», ООО «КРИПТО-ПРО», ООО «Фактор-ТС», ЗАО «МО ПНИЭИ», ООО «Инфотекс», ЗАО «СПбРЦЗИ», ООО «Криптоком», ООО «Р-Альфа». Целью этого соглашения является обеспечение взаимной совместимости продукции и решений.

Авторы выражают свою признательность

представительству компании Microsoft в России за предоставление информации о продукции и решениях компании, а также технические консультации в части PKI;

представительству RSA Security в России и компании «Демос» за активное сотрудничество и неоценимую помощь в создании этого документа;

Russ Housley (Vigil Security, LLC, housley@vigilsec.com) и Василию Сахарову (DEMOS Co Ltd, svp@dol.ru) за поощрение авторов к созданию этого документа;

Дмитрию Приходько (VSTU, PrikhodkoDV@volgablob.ru) за неоценимую помощь в корректуре этого документа т проверку формы и содержания структур ASN.1 упомянутых и использованных в этом документе.

12. Литература

12.1. Нормативные документы

[CMS] Housley, R., "Cryptographic Message Syntax (CMS)", RFC 3852, July 2004.

[CPALGS] Popov, V., Kurepkin, I., and S. Leontiev, "Additional Cryptographic Algorithms for Use with GOST 28147-89, GOST R 34.10-94, GOST R 34.10-2001, and GOST R 34.11-94 Algorithms", RFC 4357, January 2006.

[CPPK] Leontiev, S., Ed. and D. Shefanovskij, Ed., "Using the GOST R 34.10-94, GOST R 34.10-2001, and GOST R 34.11-94 Algorithms with the Internet X.509 Public Key Infrastructure Certificate and CRL Profile", RFC 4491, May 2006.

[GOST28147] "Системы обработки информации. Защита криптографическая. Алгоритм криптографического преобразования. ГОСТ 28147-89", Государственный стандарт СССР, Государственный комитет СССР по стандартизации, 1989. (на русском языке)

[GOST3431195] "Информационная технология. Криптографическая защита информации. Функция хэширования.", ГОСТ 34.311-95, Межгосударственный совет по стандартизации, метрологии и сертификации (МГС), Минск, 1995. (на русском языке)

[GOST3431095] "Информационная технология. Криптографическая защита информации. Процедуры выработки и проверки электронной цифровой подписи на базе асимметричного криптографического алгоритма.", ГОСТ 34.310-95, Межгосударственный совет по стандартизации, метрологии и сертификации (МГС), Минск, 1995. (на русском языке)

[GOST3431004] "Информационная технология. Криптографическая защита информации. Процессы формирования и проверки электронной цифровой подписи."[1], ГОСТ 34.310-2004, Межгосударственный совет по стандартизации, метрологии и сертификации (МГС), Минск, 2004. (на русском языке)

[GOSTR341094] "Информационная технология. Криптографическая защита информации. Процедуры выработки и проверки электронной цифровой подписи на базе асимметричного криптографического алгоритма.", ГОСТ Р 34.10-94, Государственный стандарт Российской Федерации, Госстандарт России, 1994. (на русском языке)

[GOSTR341001] "Информационная технология. Криптографическая защита информации. Процессы формирования и проверки электронной цифровой подписи.", ГОСТ Р 34.10-2001, Государственный стандарт Российской Федерации, Госстандарт России, 2001. (на русском языке)

[GOSTR341194] "Информационная технология. Криптографическая защита информации. Функция хэширования.", ГОСТ Р 34.11-94[2], Государственный стандарт Российской Федерации, Госстандарт России, 1994. (на русском языке)

[PROFILE] Housley, R., Polk, W., Ford, W., and D. Solo, "Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile", RFC 3280, April 2002.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC3851] Ramsdell, B., "Secure/Multipurpose Internet Mail Extensions (S/MIME) Version 3.1 Message Specification", RFC 3851, July 2004.

[X.208-88] CCITT. Recommendation X.208: Specification of Abstract Syntax Notation One (ASN.1). 1988.

[X.209-88] CCITT. Recommendation X.209: Specification of Basic Encoding Rules for Abstract Syntax Notation One (ASN.1). 1988.

12.2. Дополнительная литература

[CRYPTOLIC] «Положение о лицензировании деятельности по распространению шифровальных (криптографических) средств», Постановление Правительства РФ от 23.09.2002, N 691[3].

[RFC4134] Hoffman, P., "Examples of S/MIME Messages", RFC 4134, July 2005.

[RFEDSL] Федеральный закон «Об электронной цифровой подписи» от 10.01.2002 N 1-ФЗ[4]

[RFLLIC] Федеральный закон «О лицензировании отдельных видов деятельности» от 08.08.2001 г. N 128-ФЗ

[Schneier95] B. Schneier, Applied Cryptography, Second Edition, John Wiley & Sons, Inc., 1995.

Адреса авторов

Сергей Леонтьев, редактор
CRYPTO-PRO
38, Obraztsova,
Moscow, 127018, Russian Federation

EMail: lse@cryptopro.ru

Григорий Чудов, редактор
CRYPTO-PRO

[1] Название стандарта на английском языке в оригинале указано неверно (см. https://www.rfc-editor.org/errata/eid5099). *Прим. перев.*
[2] В оригинале ошибочно указано GOST R 31.10-94 (см. https://www.rfc-editor.org/errata/eid5089). *Прим. перев.*
[3] В соответствии с Постановлением Правительства РФ от 29 декабря 2007 г. N 957 данный документ утратил силу. *Прим. перев.*
[4] В соответствии с Федеральным законом от 6 апреля 2011 г. N 63-ФЗ утратил силу с 01.07.2013 г. *Прим. перев.*

38, Obraztsova,
Moscow, 127018, Russian Federation
EMail: chudov@cryptopro.ru

Владимир Попов
CRYPTO-PRO
38, Obraztsova,
Moscow, 127018, Russian Federation
EMail: vpopov@cryptopro.ru

Александр Афанасьев
Factor-TS
office 711, 14, Presnenskij val,
Moscow, 123557, Russian Federation
EMail: afa1@factor-ts.ru

Николай Никишин
Infotecs GmbH
p/b 35, 80-5, Leningradskij prospekt,
Moscow, 125315, Russian Federation
EMail: nikishin@infotecs.ru

Болеслав Изотов
FGUE STC "Atlas"
38, Obraztsova,
Moscow, 127018, Russian Federation
EMail: izotov@nii.voskhod.ru

Елена Минаева
MD PREI
build 3, 6A, Vtoroj Troitskij per.,
Moscow, Russian Federation
EMail: evminaeva@mail.ru

Игорь Овчаренок
MD PREI
Office 600, 14, B.Novodmitrovskaya,
Moscow, Russian Federation
EMail: igori@mo.msk.ru

Сергей Муругов
R-Alpha
4/1, Raspletina,
Moscow, 123060, Russian Federation
EMail: msm@top-cross.ru

Игорь Устинов
Cryptocom
office 239, 51, Leninskij prospekt,
Moscow, 119991, Russian Federation
EMail: igus@cryptocom.ru

Анатолий Еркин
SPRCIS (SPbRCZI)
1, Obrucheva,
St.Petersburg, 195220, Russian Federation
EMail: erkin@nevsky.net

Перевод на русский язык

Николай Малых

nmalykh@protokols.ru

Полное заявление авторских прав

Интеллектуальная собственность

IETF не принимает какой-либо позиции в отношении действительности или объема каких-либо прав интеллектуальной собственности (Intellectual Property Rights или IPR) или иных прав, которые, как может быть заявлено, относятся к реализации или использованию описанной в этом документе технологии, или степени, в которой любая лицензия, по которой права могут или не могут быть доступны, не заявляется также применение каких-либо усилий для определения таких прав. Сведения о процедурах IETF в отношении прав в документах RFC можно найти в BCP 78 и BCP 79.

Копии раскрытия IPR, предоставленные секретариату IETF, и любые гарантии доступности лицензий, а также результаты попыток получить общую лицензию или право на использование таких прав собственности разработчиками или пользователями этой спецификации, можно получить из сетевого репозитория IETF IPR по ссылке http://www.ietf.org/ipr.

IETF предлагает любой заинтересованной стороне обратить внимание на авторские права, патенты или использование патентов, а также иные права собственности, которые могут потребоваться для реализации этого стандарта. Информацию следует направлять в IETF по адресу ietf-ipr@ietf.org.

Подтверждение

Финансирование функций RFC Editor обеспечено IETF Administrative Support Activity (IASA).