ФЕДЕРАЛЬНОЕ АГЕНТСТВО ПО ТЕХНИЧЕСКОМУ РЕГУЛИРОВАНИЮ И МЕТРОЛОГИИ
(РОССТАНДАРТ)

Технический комитет 026

«Криптографическая защита информации»

Информационная технология

КРИПТОГРАФИЧЕСКАЯ ЗАЩИТА ИНФОРМАЦИИ

МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ

(ПРОЕКТ)

ИСПОЛЬЗОВАНИЕ АЛГОРИТМОВ ГОСТ 28147-89, ГОСТ Р 34.11 И ГОСТ Р 34.10 В КРИПТОГРАФИЧЕСКИХ СООБЩЕНИЯХ ФОРМАТА CMS

Проект первой редакции,
апрель 2014
rus-popov-cms-gost-00-re

Москва
2014

Введение

Настоящая рекомендация содержит описание форматов кодирования, идентификаторов и форматов параметров для алгоритмов по ГОСТ Р 34.10, ГОСТ Р 34.11 и ГОСТ 28147 при их использовании для защиты сообщений CMS в сети Интернет.

Необходимость разработки настоящей рекомендации вызвана потребностью в обеспечении совместимости использования российских алгоритмов подписи ГОСТ Р 34.10, алгоритмов функции хэширования по ГОСТ Р 34.11, алгоритмов согласования ключей VKO GOST R 34.10-2012, а также алгоритмов шифрования ГОСТ 28147 российскими производителями.

Содержание

1 Область применения

Синтаксис криптографических сообщений CMS [**IETF RFC 5652**] используется для цифровой подписи, хэширования, проверки подлинности и шифрования произвольных сообщений.

В данных рекомендациях изложены правила использования криптографических алгоритмов согласно стандартам **ГОСТ 28147-89**, **ГОСТ Р 34.10-2012** и **ГОСТ Р 34.11-2012** для сообщений CMS. В настоящем документе отсутствует описание данных криптографических алгоритмов, их определение содержится в соответствующих государственных стандартах.

Значения CMS генерируются с помощью языка ACH.1 **ГОСТ Р ИСО/МЭК 8824-1** с использованием базовых правил кодирования (BER) **ГОСТ Р ИСО/МЭК 8825-1**. В данном документе указаны идентификаторы каждого алгоритма, включая определения ACH.1 для идентификаторов объектов и всех соответствующих параметров.

Также определяются поля CMS, используемые каждым алгоритмом.

1.1 Текущий статус документа как проекта рекомендаций ТК26

Этот параграф следует удалить после принятия данного проекта рекомендаций.

Передача проекта настоящих рекомендаций в ТК26 означает, что каждый их автор соглашается с не эксклюзивным предоставлением IPR для ТК26, аналогично положениям стандарта Интернет IETF BCP 79.

Данный предварительный документ является открытым документом «Рабочей группы по сопутствующим криптографическим алгоритмам, определяющим ключевые системы» и Технического комитета по стандартизации «Криптографическая защита информации (ТК26)». Область распространения документа не ограничена.

Этот документ действителен в течении максимум девяти месяцев, и может быть в любое время изменён, заменён на другой или отозван его авторами в любое время.

При цитировании или ссылке на него из других документов следует ставить отметку «документ готовится к публикации».

Список предварительных документов ТК26 доступен по ссылке <http://www.tc26.ru/>.

Настоящий предварительный документ актуален (действителен) до января 2015 года.

2 Ссылочные документы

В настоящем документе использованы нормативные ссылки на следующие стандарты и рекомендации:

ГОСТ 28147 - «Системы обработки информации. Защита криптографическая. Алгоритм криптографического преобразования», ГОСТ 28147-89, Государственный стандарт Союза ССР, Государственный комитет СССР по стандартам, ИПК Издательство стандартов, 1996.

ГОСТ Р 34.10 - «Информационные технологии. Криптографическая защита информации. Процессы формирования и проверки электронной цифровой подписи», ГОСТ Р 34.10-2012, Национальный стандарт Российской Федерации, Федеральное агентство по техническому регулированию и метрологии, Стандартинформ, 2012.

ГОСТ Р 34.11 - «Информационные технологии. Криптографическая защита информации. Функция хэширования», ГОСТ Р 34.11-2012, Национальный стандарт Российской Федерации, Федеральное агентство по техническому регулированию и метрологии, Стандартинформ, 2012.

ГОСТ Р ИСО/МЭК 8824-1 - «Информационные технологии. Абстрактная синтаксическая нотация версии один (АСН.1). Часть 1. Спецификация основной нотации», ГОСТ Р ИСО/МЭК 8824-1-2001, Государственный стандарт Российской Федерации, Госстандарт России, Москва, 2001.

ГОСТ Р ИСО/МЭК 8825-1 - «Информационные технологии. Правила кодирования АСН.1. Часть 1. Спецификация базовых (BER), канонических (CER) и отличительных (DER) правил кодирования», ГОСТ Р ИСО/МЭК 8825-1-2003, Государственный стандарт Российской Федерации, Госстандарт России, Москва, 2003.

ТК26АЛГ - (проект) «Методические рекомендации по криптографическим алгоритмам, сопутствующим применению стандартов ГОСТ Р 34.10-2012 и ГОСТ Р 34.11-2012», документ готовится к публикации.

ТК26ИОК - (проект) «Методические рекомендации. Использование алгоритмов ГОСТ Р 34.10, ГОСТ Р 34.11 в профиле сертификата и списке отзыва сертификатов (CRL) инфраструктуры открытых ключей X.509», документ готовится к публикации.

ТК26УЗ - (проект) «Методические рекомендации по заданию узлов замены блока подстановки алгоритма шифрования ГОСТ 28147-89», документ готовится к публикации.

2.1 Дополнительные ссылки

IETF RFC 5751 - Б. Рамсделл и С. Турнер, «Спецификация сообщений для защищённых/многоцелевых расширений электронной почты (S/MIME) версии 3.2» (Ramsdell, B. and S. Turner, "Secure/Multipurpose Internet Mail Extensions (S/MIME) Version 3.2 Message Specification"), RFC 5751, январь 2010.

IETF RFC 5652 - Р. Хаусли «Синтаксис криптографических сообщений (CMS)» (Housley, R., Cryptographic Message Syntax (CMS)), RFC 5652, сентябрь 2009.

Примечание 1 - Другие международные стандарты, руководства и прочие документы по вопросам, рассматриваемым в настоящем документе, приведены в библиографии.

Примечание 2 - При пользовании настоящим документом целесообразно проверить действие ссылочных стандартов и классификаторов в информационной системе общего пользования - на официальном сайте национального органа Российской Федерации по стандартизации в сети Интернет или по ежегодно издаваемому информационному указателю «Национальные стандарты», который опубликован по состоянию на 1 января текущего года, и по соответствующим ежемесячно издаваемым информационным указателям, опубликованным в текущем году. Если ссылочный документ заменён (изменён), то при пользовании настоящим документом следует руководствоваться заменённым (изменённым) документом. Если ссылочный документ отменен без замены, то положение, в котором дана ссылка на него, применяется в части, не затрагивающей эту ссылку.

3 Определения

В настоящем документе используются следующие термины и определения, а также определяемые Федеральным законом "Об электронной подписи", №63-ФЗ от 06.04.2011:

Электронная подпись	информация в электронной форме, которая присоединена к другой информации в электронной форме (подписываемой информации) или иным образом связана с такой информацией и которая используется для определения лица, подписывающего информацию.
Сертификат ключа проверки электронной подписи	электронный документ или документ на бумажном носителе, выданные удостоверяющим центром либо доверенным лицом удостоверяющего центра и подтверждающие принадлежность ключа проверки электронной подписи владельцу сертификата ключа проверки электронной подписи.
Владелец сертификата ключа проверки электронной подписи	лицо, которому в установленном Федеральным законом «Об электронной подписи» порядке выдан сертификат ключа проверки электронной подписи.
Ключ электронной подписи	уникальная последовательность символов, предназначенная для создания электронной подписи.
Ключ проверки электронной подписи	уникальная последовательность символов, однозначно связанная с ключом электронной подписи и предназначенная для проверки подлинности электронной подписи (далее - проверка электронной подписи).
Удостоверяющий центр	юридическое лицо или индивидуальный предприниматель, осуществляющие функции по созданию и выдаче сертификатов ключей проверки электронных подписей, а также иные функции, предусмотренные Федеральным законом «Об электронной подписи».
Средства электронной подписи	шифровальные (криптографические) средства, используемые для реализации хотя бы одной из следующих функций - создание электронной подписи, проверка электронной подписи, создание ключа электронной подписи и ключа проверки электронной подписи.
Средства удостоверяющего центра	программные и (или) аппаратные средства, используемые для реализации функций удостоверяющего центра.
Участники электронного взаимодействия	осуществляющие обмен информацией в электронной форме государственные органы, органы местного самоуправления, организации, а также граждане.
Информационная система общего пользования	информационная система, участники электронного взаимодействия в которой составляют неопределённый круг лиц и в использовании которой этим лицам не может быть отказано.

4 Алгоритмы хэширования сообщений ГОСТ Р 34.11

В данном разделе изложены правила использования алгоритмов хэширования по ГОСТ Р 34.11, применяемого в CMS.

Хэш-код указывается в поле *digest* структуры *DigestedData* и в подписанном атрибуте хэш-кода сообщения (*MessageDigest*). Кроме того, хэш-код является входным параметром для алгоритмов подписей.

4.1 Алгоритм ГОСТ Р 34.11-2012 с длиной хэш-кода 256 бит

Алгоритм хэширования по ГОСТ Р 34.11-2012 с длиной 256 имеет следующий идентификатор:

```
id-tc26-gost3411-2012-256 OBJECT IDENTIFIER ::=
    { iso(1) member-body(2) ru(643) rosstandart(7) tc26(1)
        algorithms (1) digest(2) gost3411-2012-256 (2) }
```

В структуре *AlgorithmIdentifier* ДОЛЖНО присутствовать поле *parameters*, и оно ДОЛЖНО содержать значение NULL.

При наличии подписанного атрибута хэш-значение сообщения DigestedData содержит 32-байтный хэш-код:

```
GostR3411-2012-256-Digest ::= OCTET STRING (SIZE (32))
```

4.2 Алгоритм ГОСТ Р 34.11-2012 с длиной хэш-кода 512 бит

Алгоритм хэширования по ГОСТ Р 34.11-2012 с длиной 512 имеет следующий идентификатор:

```
id-tc26-gost3411-2012-512 OBJECT IDENTIFIER ::=
    { iso(1) member-body(2) ru(643) rosstandart(7) tc26(1)
        algorithms (1) digest(2) gost3411-2012-512(3) }
```

В структуре *AlgorithmIdentifier* ДОЛЖНО присутствовать поле *parameters*, и оно ДОЛЖНО содержать значение NULL.

При наличии подписанного атрибута хэш-значение сообщения DigestedData содержит 64-байтный хэш-код:

```
GostR3411-2012-512-Digest ::= OCTET STRING (SIZE (64))
```

5 Алгоритмы подписи согласно ГОСТ Р 34.10

В данном разделе описано использование алгоритмов подписи по ГОСТ Р 34.10 в CMS.

Идентификаторы алгоритма подписи указываются в поле *signatureAlgorithm* структуры *SignerInfo*, вложенной в структуру *SignedData*. *И*дентификаторы алгоритма подписи также указываются в поле *signatureAlgorithm* структуры *SignerInfo* атрибутов удостоверяющей подписи.

Значения подписи указываются в поле *signature* структуры *SignerInfo*, вложенной в структуру *SignedData*. Значения подписи также указываются в поле подписи *SignerInfo* атрибутов удостоверяющей подписи.

5.1 Алгоритм ГОСТ Р 34.10-2012 с ключом 256 бит

Алгоритм подписи по ГОСТ Р 34.10-2012 с ключом 256 бит имеет следующий идентификатор алгоритма открытого ключа:

```
id-tc26-gost3410-2012-256-signature OBJECT IDENTIFIER ::=
    id-tc26-gost3410-2012-256
```

Параметр id-tc26-gost3410-2012-256 определяется документом [**ТК26ИОК**].

Алгоритм подписи по ГОСТ Р 34.10-2012 с ключом 256 бит используется для формирования цифровой подписи в форме двух 256-битных чисел, r и s по хэш-коду ГОСТ Р 34.11-2012 с длиной хэш-кода 256 бит. Её представление в виде строки октетов (OCTET STRING) идентично представлению подписи ГОСТ Р 34.10-2001 [**IETF RFC 4490**] и состоит из 64 октетов; при этом первые 32 октета содержат число *s* в представлении big-endian (старший октет записывается первым), а вторые 32 октета содержат число *r* в представлении big-endian.

```
GostR3410-2012-256-Signature ::= OCTET STRING (SIZE (64))
```

5.2 Алгоритм ГОСТ Р 34.10-2012 с ключом 512 бит

Алгоритм подписи по ГОСТ Р 34.10-2012 с ключом 512 бит имеет следующий идентификатор алгоритма открытого ключа:

```
id-tc26-gost3410-2012-512-signature OBJECT IDENTIFIER ::=
    id-tc26-gost3410-2012-512
```

Параметр id-tc26-gost3410-2012-512 определяется документом [**ТК26ИОК**].

Алгоритм подписи по ГОСТ Р 34.10-2012 с ключом 512 бит используется для формирования цифровой подписи в форме двух 256-битных чисел, r и s по хэш-значению ГОСТ Р 34.11-2012 с длиной хэш-кода 512 бит. Её представление в виде строки октетов (OCTET STRING) состоит из 128 октетов; при этом первые 64 октета содержат число *s* в представлении big-endian (старший октет записывается первым), а вторые 64 октета содержат число *r* в представлении big-endian.

```
GostR3410-2012-512-Signature ::= OCTET STRING (SIZE (128))
```

6 Алгоритмы управления ключами

В настоящей главе описываются алгоритмы согласования и передачи ключей, основанные на алгоритме создания производных ключей VKO GOST R 34.10-2012 [**ТК26АЛГ**] и алгоритмах шифрования ключей, смотри Приложение А настоящего документа.

6.1 Алгоритмы согласования ключей

Идентификаторы алгоритма согласования ключей указываются в полях *EnvelopedData RecipientInfos KeyAgreeRecipientInfo keyEncryptionAlgorithm* и *AuthenticatedData RecipientInfos KeyAgreeRecipientInfo keyEncryptionAlgorithm*

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

6.1.1 Алгоритм согласования ключей на основе открытых ключей по ГОСТ Р 34.10

Поле *EnvelopedData RecipientInfos KeyAgreeRecipientInfo* используется следующим образом:

- В поле *originator* ДОЛЖНО быть указано значение *originatorKey*.
- Поле алгоритма originatorKey ДОЛЖНО содержать идентификатор открытого ключа ГОСТ Р 34.10 и соответствующие параметры [**ТК26ИОК**].
- Поле *originatorKey publicKey* ДОЛЖНО содержать открытый ключ отправителя.
- В качестве *keyEncryptionAlgorithm* ДОЛЖЕН быть указан идентификатор алгоритма id-tc26-agreement-gost-3410-12-256 или id-tc26-agreement-gost-3410-2012-512. Полем параметра идентификатора для данного алгоритма является поле *KeyWrapAlgorithm*, данный параметр ДОЛЖЕН быть указан. *KeyWrapAlgorithm* обозначает алгоритм и параметры, используемые для шифрования ключа, шифрующего содержимое, с помощью парного ключа для шифрования ключей, сгенерированного с помощью алгоритма согласования ключей VKO GOST R 34.10-2012.
- Синтаксис идентификаторов и параметров алгоритма выглядит следующим образом:

```
id-tc26-agreement-gost-3410-12-256  OBJECT IDENTIFIER ::=
    { iso(1) member-body(2) ru(643) rosstandart(7) tc26(1)
        algorithms (1) agreement(6) gost3410-2012-256(1) }

KeyWrapAlgorithm ::= AlgorithmIdentifier
```

ДОЛЖНЫ присутствовать параметры алгоритма *KeyWrapAlgorithm*. Синтаксис параметров алгоритма *KeyWrapAlgorithm* выглядит следующим образом:

```
Gost28147-89-KeyWrapParameters ::=
    SEQUENCE {
        encryptionParamSet Gost28147-89-ParamSet,
        ukm                OCTET STRING (SIZE (8..16)) OPTIONAL
    }

Gost28147-89-ParamSet ::= OBJECT IDENTIFIER
```

Ключевой материал пользователя (UKM) Gost28147-89-KeyWrapParameters ДОЛЖЕН отсутствовать.

Ключевой материал пользователя (UKM) *KeyAgreeRecipientInfo* ДОЛЖЕН присутствовать и содержать восемь или шестнадцать октетов для VKO GOST R 34.10-2012.

Поле encryptedKey ДОЛЖНО заключать в себе структуру Gost28147-89-EncryptedKey, где *maskKey* ДОЛЖЕН отсутствовать.

```
Gost28147-89-EncryptedKey ::=
    SEQUENCE {
        encryptedKey         Gost28147-89-Key,
        maskKey              [0] IMPLICIT Gost28147-89-Key
```

```
                                    OPTIONAL,
        macKey                  Gost28147-89-MAC
    }
```

Для формирования ключа шифрования ключей (КЕК) с помощью закрытого ключа, соответствующего ключу *originatorKey publicKey* и открытому ключу получателя, применяется алгоритм VKO GOST R 34.10-2012.

Затем алгоритм шифрования ключей, указанный в *KeyWrapAlgorithm*, применяется для формирования CEK_ENC, CEK_MAC и UKM. Для всех операций шифрования ключей используются параметры encryptionParamSet структуры Gost28147-89-KeyWrapParameters. Рекомендуется согласовывать их равными полю encryptionParamSet открытого ключа получателя.

Полученный зашифрованный ключ (CEK_ENC) помещается в поле Gost28147-89-EncryptedKey encryptedKey, имитовставка выработанная на него (CEK_MAC) помещается в поле Gost28147-89-EncryptedKey macKey, а ключевой материал пользователя (UKM) – в поле *KeyAgreeRecipientInfo ukm*.

6.2 Алгоритмы передачи ключей

В данном разделе изложены соглашения, используемые при реализации CMS с поддержкой передачи ключей с помощью алгоритма, описанного в [**ТК26АЛГ**].

Идентификаторы алгоритма передачи ключей указаны в поле *EnvelopedData RecipientInfos KeyTransRecipientInfo keyEncryptionAlgorithm*.

Ключи шифрования содержимого, зашифрованные на ключе передачи, указаны в поле *EnvelopedData RecipientInfos KeyTransRecipientInfo encryptedKey*.

6.2.1 Алгоритмы передачи ключей на основе открытых ключей по ГОСТ Р 34.10

Поле *EnvelopedData RecipientInfos KeyTransRecipientInfo* используется следующим образом:

- *keyEncryptionAlgorithm* и параметры ДОЛЖНЫ совпадать с алгоритмом и параметрами открытого ключа получателя.
- *encryptedKey* заключает в себе структуру GostR3410-KeyTransport, состоящую из зашифрованного ключа шифрования содержимого, его кода аутентификации сообщения (MAC), параметров алгоритма по ГОСТ 28147-89, используемых для шифрования ключа, эфемерного открытого ключа отправителя и ключевого материала пользователя (UKM) (*UserKeyingMaterial*; [**IETF RFC 5652**]).
- Параметры *transportParameters* ДОЛЖНЫ присутствовать.
- Ключ *ephemeralPublicKey* ДОЛЖЕН 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
    }
```

Для формирования ключа шифрования ключей (КЕК) с помощью закрытого ключа, соответствующего ключу GostR3410-TransportParameters ephemeralPublicKey, и открытого ключа получателя, применяется алгоритм VKO GOST R 34.10-2012.

Затем алгоритм шифрования ключей, смотри Приложение А настоящего документа, применяется для формирования CEK_ENC, CEK_MAC и UKM. Для всех операций шифрования ключей используются параметры encryptionParamSet структуры GostR3410-TransportParameters.

Рекомендуется согласовывать их равными полю encryptionParamSet открытого ключа получателя.

Полученный зашифрованный ключ (CEK_ENC) помещается в поле Gost28147-89-EncryptedKey encryptedKey, имитовставка выработанная на него (CEK_MAC) помещается в поле Gost28147-89-EncryptedKey macKey, а ключевой материал пользователя (UKM) – в поле GostR3410-TransportParameters ukm.

7 Алгоритмы шифрования содержимого

В данном разделе изложены соглашения, используемые при реализации CMS с поддержкой шифрования содержимого согласно ГОСТ 28147-89.

Идентификаторы алгоритма шифрования содержимого указываются в полях *EnvelopedData EncryptedContentInfo contentEncryptionAlgorithm* и *EncryptedData EncryptedContentInfo contentEncryptionAlgorithm*.

Алгоритмы шифрования содержимого используются для шифрования содержимого, указанного в полях *EnvelopedData EncryptedContentInfo encryptedContent* и *EncryptedData EncryptedContentInfo encryptedContent*.

7.1 Алгоритм шифрования содержимого по ГОСТ 28147-89

В данном разделе описывается использование алгоритма по ГОСТ 28147-89 для шифрования данных.

В настоящем документе для данного алгоритма указан следующий 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 иметь следующую структуру:

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

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

encryptionParamSet определяет согласованные параметры алгоритма шифрования, используется режим гаммирования с обратной связью и алгоритм усложнения ключей, смотри Приложение Б настоящего документа.

Согласование параметров алгоритма шифрования содержимого между получателем и отправителем сообщения может быть обеспечено использованием атрибута SMIMECapabilites в составе расширения сертификата X.509, передачей подписанных сообщений с таким атрибутом или иным способом.

8 Алгоритмы вычисления кода аутентификации сообщения

В данном разделе изложены соглашения, используемые при реализации CMS с поддержкой кода аутентификации сообщения согласно ГОСТ 34.11.

Идентификаторы алгоритма указываются в поле *AuthenticatedData macAlgorithm*.

Значения кода аутентификации указываются в поле *AuthenticatedData mac*.

8.1 Алгоритм аутентификации сообщения на основе функции хэширования ГОСТ Р 34.11-2012 с длиной хэш-кода 256 бит

Функция HMAC_GOSTR3411_2012_256(K,text) основана на функции хэширования по ГОСТ Р 34.11-2012 с длиной хэш-кода 256 бит [**ТК26АЛГ**].

В настоящем документе для данного алгоритма указан следующий идентификатор объекта (OID):

```
id-tc26-hmac-gost-3411-2012-256 OBJECT IDENTIFIER ::=
    { iso(1) member-body(2) ru(643) rosstandart(7) tc26(1)
        algorithms (1) mac(4) gost3411-2012-256(1) }
```

В структуре *AlgorithmIdentifier* ДОЛЖНО присутствовать поле *parameters*, и оно ДОЛЖНО содержать значение NULL.

8.2 Алгоритм аутентификации сообщения на основе функции хэширования ГОСТ Р 34.11-2012 с длиной хэш-кода 512 бит

Функция HMAC_GOSTR3411_2012_512(K,text) основана на функции хэширования по ГОСТ Р 34.11-2012 с длиной хэш-кода 512 бит [**ТК26АЛГ**].

В настоящем документе для данного алгоритма указан следующий идентификатор объекта (OID):

```
id-tc26-hmac-gost-3411-2012-512 OBJECT IDENTIFIER ::=
    { iso(1) member-body(2) ru(643) rosstandart(7) tc26(1)
        algorithms (1) mac(4) gost3411-2012-512(2) }
```

В структуре *AlgorithmIdentifier* ДОЛЖНО присутствовать поле *parameters*, и оно ДОЛЖНО содержать значение NULL.

9 Использование формата S/MIME

В данном разделе описывается применение алгоритмов, определённых в настоящем документе, в сообщениях формата S/MIME [**IETF RFC 5751**].

9.1 Параметр micalg

При использовании алгоритмов, определённых в настоящем документе, параметр micalg СЛЕДУЕТ установить в одно из следующих значениий: "gostr3411-2012-256" или "gostr3411-2012-512". В ином случае ДОЛЖНО быть указано значение «unknown» («неизвестно»).

9.2 Атрибут SMIMECapabilities

Значение *SMIMECapability*, указывающее на поддержку алгоритма хэширования по ГОСТ Р 34.11-2012 с длинной хэш-кода 256 бит, является последовательностью (SEQUENCE) с полем *capabilityID*, которое содержит идентификатор объекта id-tc26-gost3411-2012-256 и не содержит параметры.

Данное значение при кодировании DER выглядит следующим образом:

```
30 0A 06 08  2A 85 03 07  01 01 01 01
```

Значение *SMIMECapability*, указывающее на поддержку алгоритма хэширования по ГОСТ Р 34.11-2012 с длинной хэш-кода 512 бит, является последовательностью (SEQUENCE) с полем *capabilityID*, которое содержит идентификатор объекта id-tc26-gost3411-2012-512 и не содержит параметры.

Данное значение при кодировании DER выглядит следующим образом:

```
30 0A 06 08  2A 85 03 07  01 01 01 02
```

Значение *SMIMECapability*, указывающее на поддержку алгоритма шифрования по ГОСТ 28147-89, является последовательностью (SEQUENCE) с полем *capabilityID*, которое содержит идентификатор объекта id-Gost28147-89 и не содержит параметры. Данное значение при кодировании DER выглядит следующим образом:

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

Если отправитель желает указать поддержку определённого набора параметров, поле parameters в *SMIMECapability* ДОЛЖНО содержать структуру Gost28147-89-Parameters. Получатели ДОЛЖНЫ игнорировать поле Gost28147-89-Parameters iv и полагать, что отправитель поддерживает параметры, указанные в поле Gost28147-89-Parameters encryptionParamSet.

Структура *SMIMECapability*, указывающая на поддержку ГОСТ 28147-89 с набором параметров id-tc26-gost-28147-param-Z [**ТК26УЗ**], при кодировании DER выглядит следующим образом:

```
30 1F 06 06  2A 85 03 02  02 15 30 15  04 08 00 00
00 00 00 00  00 00 06 09  2A 85 03 07  01 02 05 01
01
```

10 Вопросы безопасности

Приложения, совместимые с настоящим документом, ДОЛЖНЫ использовать уникальные значения ukm и iv.

Получатели МОГУТ проверять, являются ли указанные отправителем значения ukm и iv уникальными.

Приложениям РЕКОМЕНДУЕТСЯ проверять значения подписей, открытые ключи и параметры алгоритмов перед использованием на предмет их соответствия стандартам ГОСТ Р 34.10.

11 Требования по совместимости

Требования по реализации CMS на основе ГОСТ 28147-89, ГОСТ Р 34.11 И ГОСТ Р 34.10:

- поддержка ГОСТ Р 34.11-2012 и ГОСТ Р 34.10-2012 со значением длины хэш-кода 256 бит – обязательно;
- id-GostR3410-2001-CryptoPro-XchA-ParamSet — обязательно [**ТК26ИОК**];
- id-tc26-gost-3410-12-512-paramSetA — при поддерже ГОСТ Р 34.11-2012 и ГОСТ Р 34.10-2012 со значением длины хэш-кода 512 бит [**ТК26ЭК**];
- id-tc26-gost-28147-param-Z — при поддержке зашифрованных сообщений [**ТК26УЗ**].

Приложение ААлгоритмы шифрования ключей (нормативное)

Алгоритмы шифрования ключа в режиме простой замены и шифрования ключа с диверсификацией идентичны алгоритмам GOST 28147-89 Key Wrap и CryptoPro Key Wrap [**IETF RFC 4357**].

A.1. **Шифрование ключа в режиме простой замены**

Идентификатор алгоритма:

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

Алгоритм шифрует ключ шифрования содержимого (CEK) размера 256 бит с использованием ключа шифрования ключей ГОСТ 28147 (KEK):

1) В качестве UKM принять 8 уникальных одноразовых (случайных) октетов. Для KEK, полученного алгоритмом VKO GOST R 34.10-2012, допустимо использовать первые 8 октетов UKM, который был использован при согласовании ключей;
2) С использованием ключа KEK выработать имитовставку CEK_MAC размера 4 октета на значение CEK:
   CEK_MAC = gost28147IMIT(UKM, KEK, CEK);
3) В качестве CEK_ENC принять значение CEK зашифрованное в режиме простой замены на ключе KEK;
4) Зашифрованный ключ является последовательностью:
   (UKM | CEK_ENC | CEK_MAC);

A.2. **Шифрование ключа с диверсификацией**

Идентификатор алгоритма:

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

Алгоритм шифрует ключ шифрования содержимого (CEK) размера 256 бит с использованием диверсифицированного ключа шифрования ключей ГОСТ 28147 (KEK):

1) В качестве UKM принять 8 уникальных одноразовых (случайных) октетов. Для KEK, полученного алгоритмом VKO GOST R 34.10-2012, допустимо использовать первые 8 октетов UKM, который был использован при согласовании ключей;
2) Получить диверсифицированный ключ KEK(UKM) по алгоритму описанному ниже;
3) С использованием ключа KEK(UKM) выработать имитовставку CEK_MAC размера 4 октета на значение CEK:
   CEK_MAC = gost28147IMIT(UKM, KEK(UKM), CEK).
4) В качестве CEK_ENC принять значение CEK зашифрованное в режиме простой замены на ключе KEK.
5) Зашифрованный ключ является последовательностью:
   (UKM | CEK_ENC | CEK_MAC).

Алгоритм диверсификации ключа, для данного ключа размера 256 бит и уникального одноразового (случайного) UKM алгоритм вырабатывает новый ключ K(UKM) (алгоритм идентичен CryptoPro KEK Diversification Algorithm [**IETF RFC 4357**]):

1) Пусть K[0] = K;
2) Разбить UKM на компоненты a[i,j] :
   UKM = a[0]|...|a[7] (a[i] - октеты, a[i,0]..a[i,7] - биты)
3) Пусть i = 0.
4) Вычислить K[1]..K[8] повторением следующего алгоритма 8 раз:
   A) Разбить K[i] на компонентв k[i,j]:
      K[i] = k[i,0]|k[i,1]|..|k[i,7] (k[i,j] - 32-бит целые)
   B) Вычислить векторS[i]:
      S[i] = ((a[i,0]*k[i,0] + ... + a[i,7]*k[i,7]) mod 2^32) |
         (((~a[i,0])*k[i,0] + ... + (~a[i,7])*k[i,7]) mod 2^32);
   C) K[i+1] = encryptCFB (S[i], K[i], K[i])
   D) i = i + 1
5) В качестве результата K(UKM) взять K[8].

Приложение БАлгоритм усложнения ключа (нормативное)

Алгоритм усложнения ключа идентичен алгоритму CryptoPro Key Meshing [**IETF RFC 4357**]. Алгоритм преобразует ключ и состояние шифратора каждые 1024 октета (8192 бита) открытого текста.

Идентификатор алгоритма:

```
id-Gost28147-89-CryptoPro-KeyMeshing  OBJECT IDENTIFIER ::=
    { iso(1) member-body(2) ru(643) rans(2) cryptopro(2)
        keyMeshing(14) cryptoPro(1) }
```

Параметры алгоритма не предусмотрены, если идентификатор используется в структуре AlgorithmIdentifier, то поле *parameters* ДОЛЖНО присутствовать, и оно ДОЛЖНО содержать значение NULL.

Режимы гаммирования, гаммирования с обратной связью и выработки имитовставки первоначально используют ключ K[0] = K, внутреннее состояние STATE0[0] и индекс i = 0. В режиме гаммирования внутренним состоянием являются регистры N3 и N4, и STATE0[0] = encryptECB(K, IV). В режиме гаммирования с обратной связью внутренним состоянием являются регистры N1 и N2, и STATE0[0] = IV. В режиме выработки имитовставки алгоритм усложнения ключа не изменяет регистры шифратора, выполняются только преобразования самого ключа, описанные ниже.

Пусть STATEn[0] – это значение состояние шифратора после обработки первых 1024 октетов данных. Обработка следующих 1024 октетов начинается с K[1] и STATE0[1], которые вычисляются следующим образом:

```
    K[i+1] = decryptECB (K[i], C);
    STATE0[i+1] = encryptECB (K[i+1], STATEn[i])

Where C = {0x69, 0x00, 0x72, 0x22,   0x64, 0xC9, 0x04, 0x23,
           0x8D, 0x3A, 0xDB, 0x96,   0x46, 0xE9, 0x2A, 0xC4,
           0x18, 0xFE, 0xAC, 0x94,   0x00, 0xED, 0x07, 0x12,
           0xC0, 0x86, 0xDC, 0xC2,   0xEF, 0x4C, 0xA9, 0x2B};
```

После обработки каждых 1024 октетов данных:

- Полученное состояние сохраняется как STATEn[i];
- Вычисляются K[i+1] и STATE0[i+1];
- i увеличивается;
- Обработка следующих 1024 октетов начинается с использованием нового ключа и состояния.

Процесс повторяется до исчерпания обрабатываемых данных.

Приложение В Примеры (информативное)

Примеры сообщений получены с использованием ключей и сертификатов из примеров [**ТК26ИОК**]. Примеры даны, как в кодировке Base64 [**IETF RFC 4648**], так и в раскодированном виде АСН.1.

В.1 Сообщение с хэш-кодом ГОСТ Р 34.11-2012 (256)

Сообщение в кодировке Base64:

```
ME8GCSqGSIb3DQEHBaBCMEACAQAwDAYIKoUDBwEBAgIFADALBgkqhkiG9w0BBwEE
ID9TmiE+l8gCzCKdR0xqoyqCWjYLKpM6lJ/ZJSCNnOG7
```

АСН.1 представление сообщения:

```
0000 30      4f:  SEQUENCE {
0002 06      09:   OBJECT IDENTIFIER digestedData
000d a0      42:   [0] {
000f 30      40:    SEQUENCE {
0011 02      01:     INTEGER 0
0014 30      0c:     SEQUENCE {
0016 06      08:      OBJECT IDENTIFIER
               :       id-tc26-gost3411-2012-256
               :        (1 2 643 7 1 1 2 2)
0020 05      00:      NULL
               :      }
0022 30      0b:     SEQUENCE {
0024 06      09:      OBJECT IDENTIFIER data
               :      }
002f 04      20:     OCTET STRING
               :      3f 53 9a 21 3e 97 c8 02 cc 22 9d 47 4c 6a a3 2a
               :      82 5a 36 0b 2a 93 3a 94 9f d9 25 20 8d 9c e1 bb
               :     }
               :    }
               :   }
```

Сообщение в кодировке Base64:

```
MIGdBgkqhkiG9w0BBwWggY8wgYwCAQAwDAYIKoUDBwEBAgIFADBXBgkqhkiG9w0B
BwGgSgRI0eUg4uXy8OgsINHy8Ojh7uboIOLt8/boLCDi5f7y+iDxIOzu8P8g8fLw
5evg7Ogg7eAg9fDg4fD7/yDv6/rq+yDI4+7w5eL7BCCd0v5OkECeXah/U5dtdAWw
wMrGKPxmmnQdUAY8VX6PUA==
```

АСН.1 представление сообщения:

```
0000 30      9d:  SEQUENCE {
0003 06      09:   OBJECT IDENTIFIER digestedData
000e a0      8f:   [0] {
0011 30      8c:    SEQUENCE {
0014 02      01:     INTEGER 00
0017 30      0c:     SEQUENCE {
0019 06      08:      OBJECT IDENTIFIER
               :       id-tc26-gost3411-2012-256
               :        (1 2 643 7 1 1 2 2)
0023 05      00:      NULL
               :      }
0025 30      57:     SEQUENCE {
0027 06      09:      OBJECT IDENTIFIER data
0032 a0      4a:      [0] {
0034 04      48:       OCTET STRING
               :        d1 e5 20 e2 e5 f2 f0 e8 2c 20 d1 f2 f0 e8 e1 ee
               :        e6 e8 20 e2 ed f3 f6 e8 2c 20 e2 e5 fe f2 fa 20
               :        f1 20 ec ee f0 ff 20 f1 f2 f0 e5 eb e0 ec e8 20
               :        ed e0 20 f5 f0 e0 e1 f0 fb ff 20 ef eb fa ea fb
               :        20 c8 e3 ee f0 e5 e2 fb
               :       }
               :      }
```

```
007e 04      20:     OCTET STRING
               :      9d d2 fe 4e 90 40 9e 5d a8 7f 53 97 6d 74 05 b0
               :      c0 ca c6 28 fc 66 9a 74 1d 50 06 3c 55 7e 8f 50
               :     }
               :    }
               :   }
```

B.2 Сообщение с хэш-кодом ГОСТ Р 34.11-2012 (512)

Сообщение в кодировке Base64:

```
MG8GCSqGSIb3DQEHBaBiMGACAQAwDAYIKoUDBwEBAgMFADALBgkqhkiG9w0BBwEE
QI6UXaIJqoafBFWShSm8rkZ56Yc6twe1UxX1bOuYvvCnNi9xVSg1bug82l8qrExq
0ro6cVwbzYHLjp+Qv0wcGoo=
```

ACH.1 представление сообщения:

```
0000 30      6f:  SEQUENCE {
0002 06      09:   OBJECT IDENTIFIER digestedData
000d a0      62:    [0] {
000f 30      60:    SEQUENCE {
0011 02      01:     INTEGER 0
0014 30      0c:     SEQUENCE {
0016 06      08:      OBJECT IDENTIFIER
               :       id-tc26-gost3411-2012-512
               :        (1 2 643 7 1 1 2 3)
0020 05      00:      NULL
               :      }
0022 30      0b:     SEQUENCE {
0024 06      09:      OBJECT IDENTIFIER data
               :      }
002f 04      40:     OCTET STRING
               :      8e 94 5d a2 09 aa 86 9f 04 55 92 85 29 bc ae 46
               :      79 e9 87 3a b7 07 b5 53 15 f5 6c eb 98 be f0 a7
               :      36 2f 71 55 28 35 6e e8 3c da 5f 2a ac 4c 6a d2
               :      ba 3a 71 5c 1b cd 81 cb 8e 9f 90 bf 4c 1c 1a 8a
               :     }
               :    }
               :   }
```

Сообщение в кодировке Base64:

```
MIG0BgkqhkiG9w0BBwGgaYwgaMCAQAwDAYIKoUDBwEBAgMFADBOBgkqhkiG9w0B
BwGgQQQ/MDEyMzQ1Njc4OTAxMjM0NTY3ODkwMTIzNDU2Nzg5MDEyMzQ1Njc4OTAx
MjM0NTY3ODkwMTIzNDU2Nzg5MDEyBEAbVNAaSvW51cw9htaNKFRisZq8JHUiLzXA
hRIr5Lof+gCtMPh2ezqCOExldPAkwxHipIEzKwjvf0F5eJHBZG9I
```

ACH.1 представление сообщения:

```
0000 30      b4:  SEQUENCE {
0003 06      09:   OBJECT IDENTIFIER digestedData
000e a0      a6:   [0] {
0011 30      a3:    SEQUENCE {
0014 02      01:     INTEGER 0
0017 30      0c:     SEQUENCE {
0019 06      08:      OBJECT IDENTIFIER
               :       id-tc26-gost3411-2012-512
               :        (1 2 643 7 1 1 2 3)
0023 05      00:      NULL
               :      }
0025 30      4e:     SEQUENCE {
0027 06      09:      OBJECT IDENTIFIER data
0032 a0      41:      [0] {
0034 04      3f:       OCTET STRING
               :        30 31 32 33 34 35 36 37 38 39 30 31 32 33 34 35
               :        36 37 38 39 30 31 32 33 34 35 36 37 38 39 30 31
                        32 33 34 35 36 37 38 39 30 31 32 33 34 35 36 37
                        38 39 30 31 32 33 34 35 36 37 38 39 30 31 32
               :       }
               :      }
0075 04      40:     OCTET STRING
```

```
                :       1b 54 d0 1a 4a f5 b9 d5 cc 3d 86 d6 8d 28 54 62
                :       b1 9a bc 24 75 22 2f 35 c0 85 12 2b e4 ba 1f fa
                :       00 ad 30 f8 76 7b 3a 82 38 4c 65 74 f0 24 c3 11
                :       e2 a4 81 33 2b 08 ef 7f 41 79 78 91 c1 64 6f 48
                :     }
                :    }
                :   }
```

B.3 Подписанное сообщение по ГОСТ Р 34.10-2012 (256)

Сообщение в кодировке Base64:

```
MIIBBQYJKoZIhvcNAQcCoIH3MIH0AgEBMQ4wDAYIKoUDBwEBAgIFADAbBgkqhkiG
9w0BBwGgDgQMVGVzdCBtZXNzYWdlMYHBMIG+AgEBMFswVjEpMCcGCSqGSIb3DQEJ
ARYaR29zdFIzNDEwLTIwMTJAZXhhbXBsZS5jb20xKTAnBgNVBAMTIEdvc3RSMzQx
MC0yMDEyICgyNTYgYml0KSBleGFtcGxlAgEBMAwGCCqFAwcBAQICBQAwDAYIKoUD
BwEBAQEFAARAkptb2ekZbC94FaGDQeP70ExvTkXtOY9zgz3cCco/hxPhXUVo3eCx
VNwDQ8enFItJZ8DEX4blZ8QtziNCMl5HbA==
```

ACH.1 представление сообщения:

```
0000 30   01 05:  SEQUENCE {
0004 06      09:   OBJECT IDENTIFIER signedData
000f a0      f7:   [0] {
0012 30      f4:    SEQUENCE {
0015 02      01:     INTEGER 1
0018 31      0e:     SET {
001a 30      0c:      SEQUENCE {
001c 06      08:       OBJECT IDENTIFIER
                :        id-tc26-gost3411-2012-256
                :         (1 2 643 7 1 1 2 2)
0026 05      00:       NULL
                :       }
                :      }
0028 30      1b:     SEQUENCE {
002a 06      09:      OBJECT IDENTIFIER data
0035 a0      0e:      [0] {
0037 04      0c:       OCTET STRING
                :        54 65 73 74 20 6d 65 73 73 61 67 65
                :       }
                :      }
0045 31      c1:     SET {
0048 30      be:      SEQUENCE {
004b 02      01:       INTEGER 1
004e 30      5b:       SEQUENCE {
0050 30      56:        SEQUENCE {
0052 31      29:         SET {
0054 30      27:          SEQUENCE {
0056 06      09:    OBJECT IDENTIFIER emailAddress (1 2 840 113549 1 9 1)
0061 16      1a:           IA5 STRING 'GostR3410-2012@example.com'
                :          }
                :         }
007d 31      29:         SET {
007f 30      27:          SEQUENCE {
0081 06      03:           OBJECT IDENTIFIER commonName (2 5 4 3)
0086 13      20:           PRINTABLE STRING 'GostR3410-2012 (256 bit) example'
                :          }
                :         }
                :        }
00a8 02      01:       INTEGER 1
                :       }
00ab 30      0c:       SEQUENCE {
00ad 06      08:        OBJECT IDENTIFIER
                :         id-tc26-gost3411-2012-256
                :          (1 2 643 7 1 1 2 2)
00b7 05      00:        NULL
                :       }
00b9 30      0c:       SEQUENCE {
00bb 06      08:        OBJECT IDENTIFIER
                :         id-tc26-gost3410-2012-256
```

```
                :           (1 2 643 7 1 1 1 1)
00c5 05       00:         NULL
                :         }
00c7 04       40:        OCTET STRING
                :         92 9b 5b d9 e9 19 6c 2f 78 15 a1 83 41 e3 fb d0
                :         4c 6f 4e 45 ed 39 8f 73 83 3d dc 09 ca 3f 87 13
                :         e1 5d 45 68 dd e0 b1 54 dc 03 43 c7 a7 14 8b 49
                :         67 c0 c4 5f 86 e5 67 c4 2d ce 23 42 32 5e 47 6c
                :        }
                :       }
                :      }
                :     }
                :    }
```

B.4 Подписанное сообщение по ГОСТ Р 34.10-2012 (512)

Сообщение в кодировке Base64:

```
MIIBSQYJKoZIhvcNAQcCoIIBOjCCATYCAQExDjAMBggqhQMHAQECAwUAMBsGCSqG
SIb3DQEHAaAOBAxUZXN0IG1lc3NhZ2UxggECMIH/AgEBMFswVjEpMCcGCSqGSIb3
DQEJARYaR29zdFIzNDEwLTIwMTJAZXhhbXBsZS5jb20xKTAnBgNVBAMTIEdvc3RS
MzQxMC0yMDEyICg1MTIgYml0KSBleGFtcGxlAgEBMAwGCCqFAwcBAQIDBQAwDAYI
KoUDBwEBAQIFAASBgFyVohNhMHUi/+RAF3Gh/cC7why6v+4jPWVlx1TYlXtV8Hje
hI2Y+rP52/LO6EUHG/XcwCBbUxmRWsbUSRRBAexmaafkSdvv2FFwC8kHOcti+UPX
PS+KRYxT8vhcsBLWWxDkc1McI7aF09hqtED36mQOfACzeJjEoUjALpmJob1V
```

ACH.1 представление сообщения:

```
0000 30   01 49:  SEQUENCE {
0004 06      09:   OBJECT IDENTIFIER signedData
000f a0   01 3a:   [0] {
0013 30   01 36:    SEQUENCE {
0017 02      01:     INTEGER 1
001a 31      0e:     SET {
001c 30      0c:      SEQUENCE {
001e 06      08:       OBJECT IDENTIFIER
                :         id-tc26-gost3411-2012-512
                :           (1 2 643 7 1 1 2 3)
0028 05      00:       NULL (0 байт)
                :       }
                :      }
002a 30      1b:     SEQUENCE {
002c 06      09:      OBJECT IDENTIFIER data
0037 a0      0e:      [0] {
0039 04      0c:       OCTET STRING
                :         54 65 73 74 20 6d 65 73 73 61 67 65
                :       }
                :      }
0047 31   01 02:     SET {
004b 30      ff:      SEQUENCE {
004e 02      01:       INTEGER 1
0051 30      5b:       SEQUENCE {
0053 30      56:        SEQUENCE {
0055 31      29:         SET {
0057 30      27:          SEQUENCE {
0059 06      09:           OBJECT IDENTIFIER emailAddress (1 2 840 113549 1 9 1)
0064 16      1a:           IA5 STRING 'GostR3410-2012@example.com'
                :           }
                :          }
0080 31      29:         SET {
0082 30      27:          SEQUENCE {
0084 06      03:           OBJECT IDENTIFIER commonName (2 5 4 3)
0089 13      20:           PRINTABLE STRING 'GostR3410-2012 (512 bit) example'
                :           }
                :          }
                :         }
00ab 02      01:        INTEGER 1
                :        }
00ae 30      0c:       SEQUENCE {
00b0 06      08:        OBJECT IDENTIFIER
```

```
                :         id-tc26-gost3411-2012-512
                :          (1 2 643 7 1 1 2 3)
00ba 05       00:        NULL
                :        }
00bc 30       0c:       SEQUENCE {
00be 06       08:        OBJECT IDENTIFIER
                :         id-tc26-gost3410-2012-512
                :          (1 2 643 7 1 1 1 2)
00c8 05       00:        NULL
                :        }
00ca 04       80:       OCTET STRING
                :        5c 95 a2 13 61 30 75 22 ff e4 40 17 71 a1 fd c0
                :        bb c2 1c ba bf ee 23 3d 65 65 c7 54 d8 95 7b 55
                :        f0 78 de 84 8d 98 fa b3 f9 db f2 ce e8 45 07 1b
                :        f5 dc c0 20 5b 53 19 91 5a c6 d4 49 14 41 01 ec
                :        66 69 a7 e4 49 db ef d8 51 70 0b c9 07 39 cb 62
                :        f9 43 d7 3d 2f 8a 45 8c 53 f2 f8 5c b0 12 d6 5b
                :        10 e4 73 53 1c 23 b6 85 d3 d8 6a b4 40 f7 ea 64
                :        0e 7c 00 b3 78 98 c4 a1 48 c0 2e 99 89 a1 bd 55
                :       }
                :      }
                :     }
                :    }
                :   }
```

B.5. Создание зашифрованного сообщения с помощью согласования ключей ГОСТ Р 34.10-2012 (256)

Сообщение в кодировке Base64:

```
MIIBhgYJKoZIhvcNAQcDoIIBdzCCAXMCAQIxggEwoYIBLAIBA6BooWYwHwYIKoUD
BwEBAQEwEwYHKoUDAgIkAAYIKoUDBwEBAgIDQwAEQPAdWM4pO38iZ49UjaXQpq+a
jhTa4KwY4B9TFMK7AiYmbFKE0eX/wvu69kFMQ2o3OJTnMOlr1WHiPYOmNO6C5hOh
CgQIX+vNomZakEIwIgYIKoUDBwEBAQEwFgYHKoUDAgINADALBgkqhQMHAQIFAQEw
gYwwgYkwWzBWMSkwJwYJKoZIhvcNAQkBFhpHb3N0UjM0MTAtMjAxMkBleGFtcGxl
LmNvbTEpMCcGA1UEAxMgR29zdFIzNDEwLTIwMTIgMjU2IGJpdHMgZXhjaGFuZ2UC
AQEEKjAoBCCNhrZOr7x2fsjjQAeDMv/tSoNRQSSQzzxgqdnYxJ3fIAQEgYLqVDA6
BgkqhkiG9w0BBwEwEwHwYGKoUDAgIVMBUECHVmR/S+hlYiBgkqhQMHAQIFAQGADEI9
UNjyuY+54uVcHw==
```

АСН.1 представление сообщения:

```
0000 30   01 82:  SEQUENCE {
0004 06      09:   OBJECT IDENTIFIER envelopedData
000f a0   01 73:   [0] {
0013 30   01 6f:    SEQUENCE {
0017 02      01:     INTEGER 02
001a 31   01 2e:     SET {
001e a1   01 2a:      [1] {
0022 02      01:       INTEGER 03
0025 a0      68:       [0] {
0027 a1      66:        [1] {
0029 30      1f:         SEQUENCE {
002b 06      08:          OBJECT IDENTIFIER
                :           id-tc26-gost3410-2012-256
                :            (1 2 643 7 1 1 1 1)
0035 30      13:          SEQUENCE {
0037 06      07:           OBJECT IDENTIFIER
                :            id-GostR3410-2001-CryptoPro-XchA-ParamSet
                :             (1 2 643 2 2 36 0)
0040 06      08:           OBJECT IDENTIFIER
                :            id-tc26-gost3411-2012-256
                :             (1 2 643 7 1 1 2 2)
                :           }
                :          }
004a 03      43:         BIT STRING 0 unused bits, encapsulates {
004d 04      40:          OCTET STRING
                :           f0 1d 58 ce 29 3b 7f 22 67 8f 54 8d a5 d0 a6 af
                :           9a 8e 14 da e0 ac 18 e0 1f 53 14 c2 bb 02 26 26
```

```
                :           6c 52 84 d1 e5 ff c2 fb ba f6 41 4c 43 6a 37 38
                :           94 e7 30 e9 6b d5 61 e2 3d 83 a6 34 ee 82 e6 13
                :          }
                :         }
                :        }
008f a1       0a:       [1] {
0091 04       08:        OCTET STRING
                :         55 f4 9e 1c 11 72 00 8b
                :        }
009b 30       20:       SEQUENCE {
009d 06       08:        OBJECT IDENTIFIER
                :         id-tc26-gost3410-2012-256
                :          (1 2 643 7 1 1 1 1)
00a7 30       14:        SEQUENCE {
00a9 06       07:         OBJECT IDENTIFIER
                :          id-Gost28147-89-None-KeyWrap
                :           (1 2 643 2 2 13 0)
00b2 30       09:         SEQUENCE {
00b4 06       09:          OBJECT IDENTIFIER
                :           id-tc26-gost-28147-param-Z
                :            (1 2 643 7 1 2 5 1 1)
                :          }
                :         }
                :        }
00bf 30       8c:       SEQUENCE {
00c2 30       89:        SEQUENCE {
00c5 30       5b:         SEQUENCE {
00c7 30       56:          SEQUENCE {
00c9 31       29:           SET {
00cb 30       27:            SEQUENCE {
00cd 06       09:             OBJECT IDENTIFIER emailAddress (1 2 840 113549 1 9
1)
00d8 16       1a:             IA5 STRING 'GostR3410-2012@example.com'
                :            }
                :           }
00f4 31       29:           SET {
00f6 30       27:            SEQUENCE {
00f8 06       03:             OBJECT IDENTIFIER commonName (2 5 4 3)
00fd 13       20:             PRINTABLE STRING 'GostR3410-2012 256 bits exchange'
                :            }
                :           }
                :          }
011f 02       01:          INTEGER 01
                :          }
0122 04       2a:         OCTET STRING, encapsulates {
0124 30       28:          SEQUENCE {
0126 04       20:           OCTET STRING
                :            8d 86 b6 4e af bc 76 7e c8 e3 40 07 83 32 ff ed
                :            4a 83 51 41 24 90 cf 3c 60 a9 d9 d8 c4 9d df 20
0148 04       04:           OCTET STRING
                :            81 82 ea 54
                :           }
                :          }
                :         }
                :        }
                :       }
014e 30       3a:     SEQUENCE {
0150 06       09:      OBJECT IDENTIFIER data
015b 30       1f:      SEQUENCE {
015d 06       06:       OBJECT IDENTIFIER
                :        id-Gost28147-89
                :         (1 2 643 2 2 21)
0165 30       15:       SEQUENCE {
0167 04       08:        OCTET STRING
                :         75 66 47 f4 be 86 56 22
0171 06       09:        OBJECT IDENTIFIER
                :         id-tc26-gost-28147-param-Z
```

```
                :           (1 2 643 7 1 2 5 1 1)
                :         }
                :        }
017c 80      0c:       [0]
                :        42 3d 50 d8 f2 b9 8f b9 e2 e5 5c 1f
                :       }
                :      }
                :     }
                :    }
```

B.6. Создание зашифрованного сообщения с помощью согласования ключей ГОСТ Р 34.10-2012 (512)

Сообщение в кодировке Base64:

```
MIIBzAYJKoZIhvcNAQcDoIIBvTCCAbkCAQIxggF2oYIBcgIBA6CBraGBqjAhBggq
hQMHAQEBAjAVBgkqhQMHAQIBAgIGCCqFAwcBAQIDA4GEAASBgCB0nQy/Ljva/mRj
w6o+eDKIvnxwYIQB5XCHhZhCpHNZiWcFxFpYXZLWRPKifOxV7NStvqGE1+fkfhBe
btkQu0tdC1XL3LO2Cp/jX16XhW/IP5rKV84qWr1Owy/6tnSsNRb+ez6IttwVvaVV
pA6ONFy9p9gawoC8nitvAVJkWW0PoQoECDVfxzxgMTAHMCIGCCqFAwcBAQECMBYG
ByqFAwICDQAwCwYJKoUDBwECBQEBMIGMMIGJMFswVjEpMCcGCSqGSIb3DQEJARYa
R29zdFIzNDEwLTIwMTJAZXhhbXBsZS5jb20xKTAnBgNVBAMTIEdvc3RSMzQxMC0y
MDEyIDUxMiBiaXRzIGV4Y2hhbmdlAgEBBCowKAQg8C/OcxRR0Uq8nDjHrQlayFb3
WFUZEnEuAKcuG6dTOawEBLhi9hIwOgYJKoZIhvcNAQcBMB8GBiqFAwICFTAVBAiD
1wH+CX6CwgYJKoUDBwECBQEBgAzUvQI4H2zRfgNgdlY=
```

ACH.1 представление сообщения:

```
0000 30   01 cc:  SEQUENCE {
0004 06      09:   OBJECT IDENTIFIER envelopedData
000f a0   01 bd:   [0] {
0013 30   01 b9:    SEQUENCE {
0017 02      01:     INTEGER 02
001a 31   01 76:     SET {
001e a1   01 72:      [1] {
0022 02      01:       INTEGER 03
0025 a0      ad:       [0] {
0028 a1      aa:        [1] {
002b 30      21:         SEQUENCE {
002d 06      08:          OBJECT IDENTIFIER
                :           id-tc26-gost3410-2012-512
                :            (1 2 643 7 1 1 1 2)
0037 30      15:          SEQUENCE {
0039 06      09:           OBJECT IDENTIFIER
                :            id-tc26-gost-3410-12-512-paramSetB
                :             (1 2 643 7 1 2 1 2 2)
0044 06      08:           OBJECT IDENTIFIER
                :            id-tc26-gost3411-2012-512
                :             (1 2 643 7 1 1 2 3)
                :           }
                :          }
004e 03      84:         BIT STRING 0 unused bits, encapsulates {
0052 04      80:          OCTET STRING
                :           20 74 9d 0c bf 2e 3b da fe 64 63 c3 aa 3e 78 32
                :           88 be 7c 70 60 84 01 e5 70 87 85 98 42 a4 73 59
                :           89 67 05 c4 5a 58 5d 92 d6 44 f2 a2 7c ec 55 ec
                :           d4 ad be a1 84 d7 e7 e4 7e 10 5e 6e d9 10 bb 4b
                :           5d 0b 55 cb dc b3 b6 0a 9f e3 5f 5e 97 85 6f c8
                :           3f 9a ca 57 ce 2a 5a bd 4e c3 2f fa b6 74 ac 35
                :           16 fe 7b 3e 88 b6 dc 15 bd a5 55 a4 0e 8e 34 5c
                :           bd a7 d8 1a c2 80 bc 9e 2b 6f 01 52 64 59 6d 0f
                :          }
                :         }
                :        }
00d5 a1      0a:       [1] {
00d7 04      08:        OCTET STRING
                :         35 5f c7 3c 60 31 30 07
                :        }
00e1 30      22:       SEQUENCE {
```

```
00e3 06      08:        OBJECT IDENTIFIER
               :         id-tc26-gost3410-2012-512
               :          (1 2 643 7 1 1 1 2)
00ed 30      16:        SEQUENCE {
00ef 06      07:         OBJECT IDENTIFIER
               :          id-Gost28147-89-None-KeyWrap
               :           (1 2 643 2 2 13 0)
00f8 30      0b:         SEQUENCE {
00fa 06      09:          OBJECT IDENTIFIER
               :           id-tc26-gost-28147-param-Z
               :            (1 2 643 7 1 2 5 1 1)
               :          }
               :         }
               :        }
0105 30      8c:       SEQUENCE {
0108 30      89:        SEQUENCE {
010b 30      5b:         SEQUENCE {
010d 30      56:          SEQUENCE {
010f 31      29:           SET {
0111 30      27:            SEQUENCE {
0113 06      09:             OBJECT IDENTIFIER emailAddress (1 2 840 113549 1 9
1)
011e 16      1a:             IA5 STRING 'GostR3410-2012@example.com'
               :             }
               :            }
013a 31      29:           SET {
013c 30      27:            SEQUENCE {
013e 06      03:             OBJECT IDENTIFIER commonName (2 5 4 3)
0143 13      20:             PRINTABLE STRING 'GostR3410-2012 512 bits exchange'
               :             }
               :            }
               :           }
0165 02      01:           INTEGER 01
               :           }
0168 04      2a:          OCTET STRING, encapsulates {
016a 30      28:           SEQUENCE {
016c 04      20:            OCTET STRING
               :             f0 2f ce 73 14 51 d1 4a bc 9c 38 c7 ad 09 5a c8
               :             56 f7 58 55 19 12 71 2e 00 a7 2e 1b a7 53 39 ac
018e 04      04:            OCTET STRING
               :             b8 62 f6 12
               :            }
               :           }
               :          }
               :         }
               :        }
0194 30      3a:      SEQUENCE {
0196 06      09:       OBJECT IDENTIFIER data
01a1 30      1f:       SEQUENCE {
01a3 06      06:        OBJECT IDENTIFIER
               :         id-Gost28147-89
               :          (1 2 643 2 2 21)
01ab 30      15:        SEQUENCE {
01ad 04      08:         OCTET STRING
               :          83 d7 01 fe 09 7e 82 c2
01b7 06      09:         OBJECT IDENTIFIER
               :          id-tc26-gost-28147-param-Z
               :           (1 2 643 7 1 2 5 1 1)
               :         }
               :        }
01c2 80      0c:       [0]
               :        d4 bd 02 38 1f 6c d1 7e 03 60 76 56
               :       }
               :      }
               :     }
               :    }
```

B.7. Создание зашифрованного сообщения с помощью передачи ключей ГОСТ Р 34.10-2012 (256)

Сообщение в кодировке Base64:

```
MIIKGgYJKoZIhvcNAQcDoIIKCzCCCgcCAQAxggE0MIIBMAIBADBbMFYxKTAnBgkq
hkiG9w0BCQEWGkdvc3RSMzQxMC0yMDEyQGV4YW1wbGUuY29tMSkwJwYDVQQDEyBH
b3N0UjM0MTAtMjAxMiAyNTYgYml0cyBleGNoYW5nZQIBATAfBggqhQMHAQEBATAT
BgcqhQMCAiQABggqhQMHAQECAgSBrDCBqTAoBCCVJxUMdbKRzCJ5K1NWJIXnN7Ul
zaceeFlblA2qH4wZrgQEsHnIG6B9BgkqhQMHAQIFAQGgZjAfBggqhQMHAQEBATAT
BgcqhQMCAiQABggqhQMHAQECAgNDAARAFoqoLg1lV780co6GdwtjLtS4KCXv9VGR
sd7PTPHCT/5iGbvOlKNW2I8UhayJ0dv7RV7Nb1lDIxPxf4Mbp2CikgQI1b4+WpGE
sfQwggjIBgkqhkiG9w0BBwEwHwYGKoUDAgIVMBUECHYNkdvFoYdyBgkqhQMHAQIF
AQGAggiYvFFpJKILAFdXjcdLLYv4eruXzL/wOXL8y9HHIDMbSzV1GM033J5Yt/p4
H6JYe1L1hjAfE/BAAYBndof2sSUxC3/I7xj+b7M8BZ3GYPqATPtR4aCQDK6z91lx
nDBAWx0HdsStT5TOj/plMs4zJDadvIJLfjmGkt0Np8FSnSdDPOcJAO/jcwiOPopg
+Z8eIuZNmY4seegTLue+7DGqvqi1GdZdMnvXBFIKc9m5DUsC7LdyboqKImh6giZE
YZnxb8a2naersPylhrf+zp4Piwwv808yOrD6LliXUiH0RojlmuaQP4wBkb7m073h
MeAWEWSvyXzOvOOuFST/hxPEupiTRoHPUdfboJT3tNpizUhE384SrvXHpwpgivQ4
J0zF2/uzTBEupXR6dFC9rTHAK3X79SltqBNnHyIXBwe+BMqTmKTfnlPVHBUfTXZg
oakDItwKwa1MBOZeciwtUFza+7o9FZhKIandb848chGdgd5O9ksaXvPJDIPxQjZd
EBVhnXLlje4TScImwTdvYB8GsI8ljKb2bL3FjwQWGbPaOjXc2D9w+Ore8bk1E4TA
ayhypU7MH3Mq1EBZ4j0iROEFBQmYRZn8vAKZ0K7aPxcDeAnKAJxdokqrMkLgI6WX
0glh/3Cs9dI+0D2GqMSygauKCD0vTIo3atkEQswDZR4pMx88gB4gmx7iIGrc/ZXs
ZqHI7NQqeKtBwv2MCIj+/UTqdYDqbaniDwdVS8PE9nQnNU4gKffq3JbT+wRjJv6M
Dr231bQHgAsFTVKbZgoL4gj4V7bLQUmW06+W1BQUJ2+Sn7fp+Xet9Xd3cGtNdxzQ
zl6sGuiOlTNe0bfKP7QIMC7ekjflLBx8nwa2GZG19k3O0Z9JcDdN/kz6bGpPNssY
AIOkTvLQjxIM9MhRqIv6ee0rowTWQPwXJP7yHApop4XZvVX6h9gG2gazqbDej2lo
tAcfRAKj/LJ/bk9+OlNXOXVCKnwE1kXxZDsNJ51GdCungC56U/hmd3C1RhSLTpEc
FlOWgXKNjbn6SQrlq1yASKKr80T0fL7PFoYwKZoQbKMAVZQC1VBWQltHkEzdL73x
FwgZULNfdflF8sEhFC/zsVqckD/UnhzJz88PtCslMArJ7ntbEF1GzsSSfRfjBqnl
kSUreE5XX6+c9yp5HcJBiMzp6ZqqWWaED5Y5xp1hZeYjuKbDMfY4tbWVc7Hy0dD2
KGfZLp5umqvPNs7aVBPmvuxtrnxcJlUB8u2HoiHc6/TuhrpaopYGBhxL9+kezuLR
v18nsAg8HOmcCNUS46NXQj/Mdpx8W+RsyzCQkJjieT/Yed20Zxq1zJoXIS0xAaUH
TdE2dWqiT6TGlh/KQYk3KyFPNnDmzJm04a2VWIwpp4ypXyxrB7XxnVY6Q4YBYbZs
FycxGjJWqj7lwc+lgZ8YV2WJ4snEo2os8SsA2GFWcUMiVTHDnEJvphDHmhWsf26A
bbRqwaRXNjhj05DamTRsczgvfjdl1pk4lJYE4ES3nixtMe4s1X8nSmM4KvfyVDul
J8uTpw1ZFnolTdfEL63BSf4FREoEqKB7cKuD7cpn7Rg4kRdM0/BLZGuxkH+pGMsI
Bb8LecUWyjGsI6h74Wz/U2uBrfgdRqhR+UsfB2QLaRgM6kCXZ4vM0auuzBViFCwK
tYMHzZWWz8gyVtJ0mzt1DrHCMx4pTS4yOhv4RkXBS/rub4VhVIsOGOGar5ZYtH47
uBbdw3NC05JIFM7lI31d0s1fvvkTUR7eaqRW+SnR2c2oHpWlSO+Q0mrzx+vvOTdj
xa713YtklBvyUUQr2SIbsXGpFnwjn+sXK1onAavp/tEax8sNZvxg5yeseFcWn+gD
4rjk9FiSd1wp4fTDQFJ19evqruqKlq6k18l/ZAyUcEbIWSz2s3HfAAoAQyFPX1Q2
95gVhRRw6lP4S6VPCfn/f+5jV4TcT6W/giRaHIk9Hty+g8bx1bFXaKVkQZ5R2Vmk
qsZ65ZgCrYQJmcErPmYybvP7NBeDS4AOSgBQAGMQF4xywdNm6bniWWo3N/xkFv32
/25x8okGgD8QcYKmhzieLSSzOvM/exB14RO84YZOkZzm01Jll0nac/LEazKoVWbn
0VdcQ7pYEOqeMBXipsicNVYA/uhonp6op9cpIVYafPr0npCGwwhwcRuOrgSaZyCn
VG2tPkEOv9LKmUbhnaDA2YUSzOOjcCpIVvTSBnUEiorYpfRYgQLrbcd2qhVvNCLX
8ujZfMqXQXK8n5BK8JxNtczvaf+/2dfv1dQl0lHEAQhbNcsJ0t5GPhsSCC5oMBJl
ZJuOEO/8PBWKEnMZOM+Dz7gEgsBhGyMFFrKpiwQRpyEshSD2QpnK6Lp0t5C8Za2G
lhyZsEr+93AYOb5mm5+z02B4Yq9+RpepvjoqVeq/2uywZNq9MS98zVgNsmpryvTZ
3HJHHB20u2jcVu0G3Nhiv22lD70JWCYFAOupjgVcUcaBxjxwUMAvgHg7JZqs6mC6
tvTKwQ4NtDhoAhARlDeWSwCWb2vPH2H7LmqokifiRfvJ0hrLzkJuHdWrzIYzXpPs
+v9XJxLvbdKi9KU1Halq9S8dXT1fvs9DJTpUV/KW7QkRsTQJhTJBkQ07WUSJ4gBS
Qp4efxSRNIfMj7DR6qLLf13RpIPTJO9/+gNuBIFcupWVfUL7tJZt8Qsf9eGwZfP+
YyhjC8AyZjH4/9RzLHSjuq6apgw3Mzw0j572Xg6xDLMK8C3Tn/vrLOvAd96b9MkF
3+ZHSLW3IgOiy+1jvK/20CZxNWc+pey8v4zji1hI17iohsipX/uZKRxhxF6+Xn2R
UQp6qoxHAspNXgWQ57xg7C3+gmi4ciVr0fT9pg54ogcowrRH+I6wd0EpeWPbzfnQ
pRmMVN+YtRsrEHwH3ToQ/i4vrtgA+eONuKT2uKZFikxA+VNmeeGdhkgqETMihQ==
```

ACH.1 представление сообщения:

```
0000 30   0a 1a:  SEQUENCE {
0004 06      09:   OBJECT IDENTIFIER envelopedData
000f a0   0a 0b:   [0] {
0013 30   0a 07:    SEQUENCE {
0017 02      01:     INTEGER 0
001a 31   01 34:     SET {
001e 30   01 30:      SEQUENCE {
```

```
0022 02      01:       INTEGER 0
0025 30      5b:       SEQUENCE {
0027 30      56:        SEQUENCE {
0029 31      29:         SET {
002b 30      27:          SEQUENCE {
002d 06      09:           OBJECT IDENTIFIER emailAddress (1 2 840 113549 1 9 1)
0038 16      1a:           IA5 STRING 'GostR3410-2012@example.com'
         :           }
               :          }
0054 31      29:         SET {
0056 30      27:          SEQUENCE {
0058 06      03:           OBJECT IDENTIFIER commonName (2 5 4 3)
005d 13      20:           PRINTABLE STRING 'GostR3410-2012 256 bits exchange'
               :          }
               :         }
               :        }
007f 02      01:        INTEGER 1
               :        }
0082 30      1f:        SEQUENCE {
0084 06      08:         OBJECT IDENTIFIER
               :          id-tc26-gost3410-2012-256
               :           (1 2 643 7 1 1 1 1)
008e 30      13:         SEQUENCE {
0090 06      07:          OBJECT IDENTIFIER
               :           id-GostR3410-2001-CryptoPro-XchA-ParamSet
               :            (1 2 643 2 2 36 0)
0099 06      08:          OBJECT IDENTIFIER
               :           id-tc26-gost3411-2012-256
               :            (1 2 643 7 1 1 2 2)
               :          }
               :         }
00a3 04      ac: OCTET STRING, encapsulates {
00a6 30      a9:         SEQUENCE {
00a9 30      28:          SEQUENCE {
00ab 04      20:           OCTET STRING
               :            95 27 15 0c 75 b2 91 cc 22 79 2b 53 56 24 85 e7
               :            37 b5 25 cd a7 1e 78 59 5b 94 0d aa 1f 8c 19 ae
00cd 04      04:           OCTET STRING
               :            b0 79 c8 1b
               :           }
00d3 a0      7d:          [0] {
00d5 06      09:           OBJECT IDENTIFIER
               :            id-tc26-gost-28147-param-Z
               :             (1 2 643 7 1 2 5 1 1)
00e0 a0      66:           [0] {
00e2 30      1f:            SEQUENCE {
00e4 06      08:             OBJECT IDENTIFIER
               :              id-tc26-gost3410-2012-256
               :               (1 2 643 7 1 1 1 1)
00ee 30      13:             SEQUENCE {
00f0 06      07:              OBJECT IDENTIFIER
               :               id-GostR3410-2001-CryptoPro-XchA-ParamSet
               :                (1 2 643 2 2 36 0)
00f9 06      08:      OBJECT IDENTIFIER
               :               id-tc26-gost3411-2012-256
               :                (1 2 643 7 1 1 2 2)
               :             }
               :            }
0103 03      43:           BIT STRING 0 unused bits, encapsulates {
0106 04      40:            OCTET STRING
               :             16 8a a8 2e 0d 65 57 bf 34 72 8e 86 77 0b 63 2e
               :             d4 b8 28 25 ef f5 51 91 b1 de cf 4c f1 c2 4f fe
               :             62 19 bb ce 94 a3 56 d8 8f 14 85 ac 89 d1 db fb
               :             45 5e cd 6f 59 43 23 13 f1 7f 83 1b a7 60 a2 92
               :            }
               :           }
0148 04      08:           OCTET STRING
               :            d5 be 3e 5a 91 84 b1 f4
```

```
                :          }
                :         }
                :        }
                :       }
                :      }
0152 30    08 c8:     SEQUENCE {
0156 06       09:      OBJECT IDENTIFIER data
0161 30       1f:      SEQUENCE {
0163 06       06:       OBJECT IDENTIFIER
                :        id-Gost28147-89
                :         (1 2 643 2 2 21)
016b 30       15:       SEQUENCE {
016d 04       08:        OCTET STRING
                :         76 0d 91 db c5 a1 87 72
0177 06       09:        OBJECT IDENTIFIER
                :         id-tc26-gost-28147-param-Z
                :          (1 2 643 7 1 2 5 1 1)
                :        }
                :       }
0182 80    08 98:      [0]
                :       bc 51 69 24 a2 0b 00 57 57 8d c7 4b 2d 8b f8 7a
                :       bb 97 cc bf f0 39 72 fc cb d1 c7 20 33 1b 4b 35
                :       75 18 cd 37 dc 9e 58 b7 fa 78 1f a2 58 7b 52 f5
                :       86 30 1f 13 f0 40 01 80 67 76 87 f6 b1 25 31 0b
                :       7f c8 ef 18 fe 6f b3 3c 05 9d c6 60 fa 80 4c fb
                :       51 e1 a0 90 0c ae b3 f7 59 71 9c 30 40 5b 1d 07
                :       76 c4 ad 4f 94 ce 8f fa 65 32 ce 33 24 36 9d bc
                :       82 4b 7e 39 86 92 dd 0d a7 c1 52 9d 27 43 3c e7
                :       09 00 ef e3 73 08 8e 3e 8a 60 f9 9f 1e 22 e6 4d
                :       99 8e 2c 79 e8 13 2e e7 be ec 31 aa be a8 b5 19
                :       d6 5d 32 7b d7 04 52 0a 73 d9 b9 0d 4b 02 ec b7
                :       72 6e 8a 8a 22 68 7a 82 26 44 61 99 f1 6f c6 b6
                :       9d a7 ab b0 fc a5 86 b7 fe ce 9e 0f 8b 0c 2f f3
                :       4f 32 3a b0 fa 2e 58 97 52 21 f4 46 88 e5 9a e6
                :       90 3f 8c 01 91 be e6 d3 bd e1 31 e0 16 11 64 af
                :       c9 7c ce bc e3 ae 15 24 ff 87 13 c4 ba 98 93 46
                :       81 cf 51 d7 db a0 94 f7 b4 da 62 cd 48 44 df ce
                :       12 ae f5 c7 a7 0a 60 8a f4 38 27 4c c5 db fb b3
                :       4c 11 2e a5 74 7a 74 50 bd ad 31 c0 2b 75 fb f5
                :       29 6d a8 13 67 1f 22 17 07 07 be 04 ca 93 98 a4
                :       df 9e 53 d5 1c 15 1f 4d 76 60 a1 a9 03 22 dc 0a
                :       c1 ad 4c 04 e6 5e 72 2c 2d 50 5c da fb ba 3d 15
                :       98 4a 21 a9 dd 6f ce 3c 72 11 9d 81 de 4e f6 4b
                :       1a 5e f3 c9 0c 83 f1 42 36 5d 10 15 61 9d 72 e5
                :       8d ee 13 49 c2 26 c1 37 6f 60 1f 06 b0 8f 25 8c
                :       a6 f6 6c bd c5 8f 04 16 19 b3 da 3a 35 dc d8 3f
                :       70 f8 ea de f1 b9 35 13 84 c0 6b 28 72 a5 4e cc
                :       1f 73 2a d4 40 59 e2 3d 22 44 e1 05 05 09 98 45
                :       99 fc bc 02 99 d0 ae da 3f 17 03 78 09 ca 00 9c
                :       5d a2 4a ab 32 42 e0 23 a5 97 d2 09 61 ff 70 ac
                :       f5 d2 3e d0 3d 86 a8 c4 b2 81 ab 8a 08 3d 2f 4c
                :       8a 37 6a d9 04 42 cc 03 65 1e 29 33 1f 3c 80 1e
                :       20 9b 1e e2 20 6a dc fd 95 ec 66 a1 c8 ec d4 2a
                :       78 ab 41 c2 fd 8c 08 88 fe fd 44 ea 75 80 ea 6d
                :       a9 e2 0f 07 55 4b c3 c4 f6 74 27 35 4e 20 29 f7
                :       ea dc 96 d3 fb 04 63 26 fe 8c 0e bd b7 d5 b4 07
                :       80 0b 05 4d 52 9b 66 0a 0b e2 08 f8 57 b6 cb 41
                :       49 96 d3 af 96 d4 14 14 27 6f 92 9f b7 e9 f9 77
                :       ad f5 77 77 70 6b 4d 77 1c d0 ce 5e ac 1a e8 8e
                :       95 33 5e d1 b7 ca 3f b4 08 30 2e de 92 37 e5 2c
                :       1c 7c 9f 06 b6 19 91 b5 f6 4d ce d1 9f 49 70 37
                :       4d fe 4c fa 6c 6a 4f 36 cb 18 00 83 a4 4e f2 d0
                :       8f 12 0c f4 c8 51 a8 8b fa 79 ed 2b a3 04 d6 40
                :       fc 17 24 fe f2 1c 0a 68 a7 85 d9 bd 55 fa 87 d8
                :       06 da 06 b3 a9 b0 de 8f 69 68 b4 07 1f 44 02 a3
                :       fc b2 7f 6e 4f 7e 3a 53 57 39 75 42 2a 7c 04 d6
                :       45 f1 64 3b 0d 27 9d 46 74 2b a7 80 2e 7a 53 f8
                :       66 77 70 b5 46 14 8b 4e 91 1c 16 53 96 81 72 8d
```

```
:       8d b9 fa 49 0a e5 ab 5c 80 48 a2 ab f3 44 f4 7c
:       be cf 16 86 30 29 9a 10 6c a3 00 55 94 02 d5 50
:       56 42 5b 47 90 4c dd 2f bd f1 17 08 19 50 b3 5f
:       75 f9 45 f2 c1 21 14 2f f3 b1 5a 9c 90 3f d4 9e
:       1c c9 cf cf 0f b4 2b 25 30 0a c9 ee 7b 5b 10 5d
:       46 ce c4 92 7d 17 e3 06 a9 e5 91 25 2b 78 4e 57
:       5f af 9c f7 2a 79 1d c2 41 88 cc e9 e9 9a aa 59
:       66 84 0f 96 39 c6 9d 61 65 e6 23 b8 a6 c3 31 f6
:       38 b5 b5 95 73 b1 f2 d1 d0 f6 28 67 d9 2e 9e 6e
:       9a ab cf 36 ce da 54 13 e6 be ec 6d ae 7c 5c 26
:       55 01 f2 ed 87 a2 21 dc eb f4 ee 86 ba 5a a2 96
:       06 06 1c 4b f7 e9 1e ce e2 d1 bf 5f 27 b0 08 3c
:       1c e9 9c 08 d5 12 e3 a3 57 42 3f cc 76 9c 7c 5b
:       e4 6c cb 30 90 90 98 e2 79 3f d8 79 dd b4 67 1a
:       b5 cc 9a 17 21 2d 31 01 a5 07 4d d1 36 75 6a a2
:       4f a4 c6 96 1f ca 41 89 37 2b 21 4f 36 70 e6 cc
:       99 b4 e1 ad 95 58 8c 29 a7 8c a9 5f 2c 6b 07 b5
:       f1 9d 56 3a 43 86 01 61 b6 6c 17 27 31 1a 32 56
:       aa 3e e5 c1 cf a5 81 9f 18 57 65 89 e2 c9 c4 a3
:       6a 2c f1 2b 00 d8 61 56 71 43 22 55 31 c3 9c 42
:       6f a6 10 c7 9a 15 ac 7f 6e 80 6d b4 6a c1 a4 57
:       36 38 63 d3 90 da 99 34 6c 73 38 2f 7e 37 65 d6
:       99 38 94 96 04 e0 44 b7 9e 2c 6d 31 ee 2c d5 7f
:       27 4a 63 38 2a f7 f2 54 3b a5 27 cb 93 a7 0d 59
:       16 7a 25 4d d7 c4 2f ad c1 49 fe 05 44 4a 04 a8
:       a0 7b 70 ab 83 ed ca 67 ed 18 38 91 17 4c d3 f0
:       4b 64 6b b1 90 7f a9 18 cb 08 05 bf 0b 79 c5 16
:       ca 31 ac 23 a8 7b e1 6c ff 53 6b 81 ad f8 1d 46
:       a8 51 f9 4b 1f 07 64 0b 69 18 0c ea 40 97 67 8b
:       cc d1 ab ae cc 15 62 14 2c 0a b5 83 07 cd 95 96
:       cf c8 32 56 d2 74 9b 3b 75 0e b1 c2 33 1e 29 4d
:       2e 32 3a 1b f8 46 45 c1 4b fa ee 6f 85 61 54 8b
:       0e 18 e1 9a af 96 58 b4 7e 3b b8 16 dd c3 73 42
:       d3 92 48 14 ce e5 23 7d 5d d2 cd 5f be f9 13 51
:       1e de 6a a4 56 f9 29 d1 d9 cd a8 1e 95 a5 48 ef
:       90 d2 6a f3 c7 eb ef 39 37 63 c5 ae f5 dd 8b 64
:       94 1b f2 51 44 2b d9 22 1b b1 71 a9 16 7c 23 9f
:       eb 17 2b 5a 27 01 ab e9 fe d1 1a c7 cb 0d 66 fc
:       60 e7 27 ac 78 57 16 9f e8 03 e2 b8 e4 f4 58 92
:       77 5c 29 e1 f4 c3 40 52 75 f5 eb ea ae ea 8a 96
:       ae a4 d7 c9 7f 64 0c 94 70 46 c8 59 2c f6 b3 71
:       df 00 0a 00 43 21 4f 5f 54 36 f7 98 15 85 14 70
:       ea 53 f8 4b a5 4f 09 f9 ff 7f ee 63 57 84 dc 4f
:       a5 bf 82 24 5a 1c 89 3d 1e dc be 83 c6 f1 d5 b1
:       57 68 a5 64 41 9e 51 d9 59 a4 aa c6 7a e5 98 02
:       ad 84 09 99 c1 2b 3e 66 32 6e f3 fb 34 17 83 4b
:       80 0e 4a 00 50 00 63 10 17 8c 72 c1 d3 66 e9 b9
:       e2 59 6a 37 37 fc 64 16 fd f6 ff 6e 71 f2 89 06
:       80 3f 10 71 82 a6 87 38 9e 2d 24 b3 3a f3 3f 7b
:       10 75 e1 13 bc e1 86 4e 91 9c e6 d3 52 65 97 49
:       da 73 f2 c4 6b 32 a8 55 66 e7 d1 57 5c 43 ba 58
:       10 ea 9e 30 15 e2 a6 c8 9c 35 56 00 fe e8 68 9e
:       9e a8 a7 d7 29 21 56 1a 7c fa f4 9e 90 86 c3 08
:       70 71 1b 8e ae 04 9a 67 20 a7 54 6d ad 3e 41 0e
:       bf d2 ca 99 46 e1 9d a0 c0 d9 85 12 cc e3 a3 70
:       2a 48 56 f4 d2 06 75 04 8a 8a d8 a5 f4 58 81 02
:       eb 6d c7 76 aa 15 6f 34 22 d7 f2 e8 d9 7c ca 97
:       41 72 bc 9f 90 4a f0 9c 4d b5 cc ef 69 ff bf d9
:       d7 ef d5 d4 25 d2 51 c4 01 08 5b 35 cb 09 d2 de
:       46 3e 1b 12 08 2e 68 30 12 65 64 9b 8e 10 ef fc
:       3c 15 8a 12 73 19 38 cf 83 cf b8 04 82 c0 61 1b
:       23 05 16 b2 a9 8b 04 11 a7 21 2c 85 20 f6 42 99
:       ca e8 ba 74 b7 90 bc 65 ad 86 96 1c 99 b0 4a fe
:       f7 70 18 39 be 66 9b 9f b3 d3 60 78 62 af 7e 46
:       97 a9 be 3a 2a 55 ea bf da ec b0 64 da bd 31 2f
:       7c cd 58 0d b2 6a 6b ca f4 d9 dc 72 47 1c 1d b4
:       bb 68 dc 56 ed 06 dc d8 62 bf 6d a5 0f bd 09 58
:       26 05 00 eb a9 8e 05 5c 51 c6 81 c6 3c 70 50 c0
```

```
:       2f 80 78 3b 25 9a ac ea 60 ba b6 f4 ca c1 0e 0d
:       b4 38 68 02 10 11 94 37 96 4b 00 96 6f 6b cf 1f
:       61 fb 2e 6a a8 92 27 f5 45 fb c9 d2 1a cb ce 42
:       6e 1d d5 ab cc 86 33 5e 93 ec fa ff 57 27 12 ef
:       6d d2 a2 f4 a5 35 1d a9 6a f5 2f 1d 5d 3d 5f be
:       cf 43 25 3a 54 57 f2 96 ed 09 11 b1 34 09 85 32
:       41 91 0d 3b 59 44 89 e2 00 52 42 9e 1e 7f 14 91
:       34 87 cc 8f b0 d1 ea a2 cb 7f 5d d1 a4 83 d3 24
:       ef 7f fa 03 6e 04 81 5c ba 95 95 7d 42 fb b4 96
:       6d f1 0b 1f f5 e1 b0 65 f3 fe 63 28 63 0b c0 32
:       66 31 f8 ff d4 73 2c 74 a3 ba ae 9a a6 0c 37 33
:       3c 34 8f 9e f6 5e 0e b1 0c b3 0a f0 2d d3 9f fb
:       eb 2c eb c0 77 de 9b f4 c9 05 df e6 47 48 b5 b7
:       22 03 a2 cb ed 63 bc af f6 d0 26 71 35 67 3e a5
:       ec bc bf 8c e3 8b 58 48 d7 b8 a8 86 c8 a9 5f fb
:       99 29 1c 61 c4 5e be 5e 7d 91 51 0a 7a aa 8c 47
:       02 ca 4d 5e 05 90 e7 bc 60 ec 2d fe 82 68 b8 72
:       25 6b d1 f4 fd a6 0e 78 a2 07 28 c2 b4 47 f8 8e
:       b0 77 41 29 79 63 db cd f9 d0 a5 19 8c 54 df 98
:       b5 1b 2b 10 7c 07 dd 3a 10 fe 2e 2f ae d8 00 f9
:       e3 8d b8 a4 f6 b8 a6 45 8a 4c 40 f9 53 66 79 e1
:       9d 86 48 2a 11 33 22 85
:      }
:     }
:    }
:   }
```

B.8. Создание зашифрованного сообщения с помощью передачи ключей ГОСТ Р 34.10-2012 (512)

Сообщение в кодировке Base64:

```
MIIB0gYJKoZIhvcNAQcDoIIBwzCCAb8CAQAxggF8MIIBeAIBADBbMFYxKTAnBgkq
hkiG9w0BCQEWGkdvc3RSMzQxMC0yMDEyQGV4YW1wbGUuY29tMSkwJwYDVQQDEyBH
b3N0UjM0MTAtMjAxMiA1MTIgYml0cyBleGNoYW5nZQIBATAhBggqhQMHAQEBAjAV
BgkqhQMHAQIBAgIGCCqFAwcBAQIDBIHyMIHvMCgEIIsYzbVLn33aLinQ7SLNA7y+
Lrm02khqDCfXrNS9iiMhBATerS8zoIHCBgkqhQMHAQIFAQGggaowIQYIKoUDBwEB
AQIwFQYJKoUDBwECAQICBggqhQMHAQECAwOBhAAEgYAYiTVLKpSGaAvjJEDQ0hdK
qR/jek5Q9Q2pXC+NkOimQh7dpCi+wcaHlPcBk96hmpnOFvLaiokX8V6jqtBl5gdk
M40kOXv8kcDdTzEVKA/ZLxA8xanL+gTD6ZjaPsUu06nsA2MoMBWcHLUzueaP3bGT
/yHTV+Za5xdcQehag/lNBgQIvCw4uUl0XC4wOgYJKoZIhvcNAQcBMB8GBiqFAwIC
FTAVBAj+1QzaXaN9FwYJKoUDBwECBQEBgAyK54euw0sHhEVEkA0=
```

ACH.1 представление сообщения:

```
0000 30   01 d2:  SEQUENCE {
0004 06      09:   OBJECT IDENTIFIER envelopedData
000f a0   01 c3:   [0] {
0013 30   01 bf:    SEQUENCE {
0017 02      01:     INTEGER 0
001a 31   01 7c:     SET {
001e 30   01 78:      SEQUENCE {
0022 02      01:       INTEGER 0
0025 30      5b:       SEQUENCE {
0027 30      56:        SEQUENCE {
0029 31      29:         SET {
002b 30      27:          SEQUENCE {
002d 06      09:           OBJECT IDENTIFIER emailAddress (1 2 840 113549 1 9 1)
0038 16      1a:           IA5 STRING 'GostR3410-2012@example.com'
               :           }
               :          }
0054 31      29:         SET {
0056 30      27:          SEQUENCE {
0058 06      03:           OBJECT IDENTIFIER commonName (2 5 4 3)
005d 13      20:           PRINTABLE STRING 'GostR3410-2012 512 bits exchange'
               :           }
               :          }
               :         }
007f 02      01:        INTEGER 1
```

```
                :         }
0082 30       21:         SEQUENCE {
0084 06       08:          OBJECT IDENTIFIER
                :           id-tc26-gost3410-2012-512
                :            (1 2 643 7 1 1 1 2)
008e 30       15:          SEQUENCE {
0090 06       07:           OBJECT IDENTIFIER
                :            id-tc26-gost-3410-12-512-paramSetB
                :             (1 2 643 7 1 2 1 2 2)
009b 06       08:           OBJECT IDENTIFIER
                :            id-tc26-gost3411-2012-512
                :             (1 2 643 7 1 1 2 3)
                :           }
                :          }
00a5 04       f2:         OCTET STRING, encapsulates {
00a8 30       ef:          SEQUENCE {
00ab 30       28:           SEQUENCE {
00ad 04       20:            OCTET STRING
                :             8b 18 cd b5 4b 9f 7d da 2e 29 d0 ed 22 cd 03 bc
                :             be 2e b9 b4 da 48 6a 0c 27 d7 ac d4 bd 8a 23 21
00cf 04       04:            OCTET STRING
                :             de ad 2f 33
                :            }
00d5 a0       c2:           [0] {
00d8 06       09:             OBJECT IDENTIFIER
                :              id-tc26-gost-28147-param-Z
                :               (1 2 643 7 1 2 5 1 1)
00e3 a0       aa:             [0] {
00e6 30       21:              SEQUENCE {
00e8 06       08:               OBJECT IDENTIFIER
                :                id-tc26-gost3410-2012-512
                :                 (1 2 643 7 1 1 1 2)
00f2 30       15:               SEQUENCE {
00f4 06       09:                OBJECT IDENTIFIER
                :                 id-tc26-gost-3410-12-512-paramSetB
                :                  (1 2 643 7 1 2 1 2 2)
00ff 06       08:                OBJECT IDENTIFIER
                :                 id-tc26-gost3411-2012-512
                :                  (1 2 643 7 1 1 2 3)
                :                }
                :               }
0109 03       84:             BIT STRING 0 unused bits, encapsulates {
010d 04       80:              OCTET STRING
                :               18 89 35 4b 2a 94 86 68 0b e3 24 40 d0 d2 17 4a
                :               a9 1f e3 7a 4e 50 f5 0d a9 5c 2f 8d 90 e8 a6 42
                :               1e dd a4 28 be c1 c6 87 94 f7 01 93 de a1 9a 99
                :               ce 16 f2 da 8a 89 17 f1 5e a3 aa d0 65 e6 07 64
                :               33 8d 24 39 7b fc 91 c0 dd 4f 31 15 28 0f d9 2f
                :               10 3c c5 a9 cb fa 04 c3 e9 98 da 3e c5 2e d3 a9
                :               ec 03 63 28 30 15 9c 1c b5 33 b9 e6 8f dd b1 93
                :               ff 21 d3 57 e6 5a e7 17 5c 41 e8 5a 83 f9 4d 06
                :              }
                :             }
0190 04       08:            OCTET STRING
                :             bc 2c 38 b9 49 74 5c 2e
                :            }
                :           }
                :          }
                :         }
                :        }
019a 30       3a:      SEQUENCE {
019c 06       09:       OBJECT IDENTIFIER data
01a7 30       1f:       SEQUENCE {
01a9 06       06:        OBJECT IDENTIFIER
                :         id-Gost28147-89
                :          (1 2 643 2 2 21)
01b1 30       15:        SEQUENCE {
01b3 04       08:         OCTET STRING
```

```
                :          fe d5 0c da 5d a3 7d 17
01bd 06       09:         OBJECT IDENTIFIER
                :          id-tc26-gost-28147-param-Z
                :            (1 2 643 7 1 2 5 1 1)
                :          }
                :        }
01c8 80       0c:       [0]
                :        8a e7 87 ae c3 4b 07 84 45 44 90 0d
                :       }
                :     }
                :    }
                :   }
```

Библиография

[**IETF RFC 4357**] В. Попов, И. Курепкин и С. Леонтьев, «Дополнительные алгоритмы шифрования для использования с алгоритмами по ГОСТ 28147-89, ГОСТ Р 34.10-94, ГОСТ Р 34.10-2001 и ГОСТ Р 34.11-94» (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, январь 2006 г.

[**IETF RFC 4490**] Под ред. С. Леонтьева и Г. Чудова «Использование алгоритмов ГОСТ 28147-89, ГОСТ Р 34.11-94, ГОСТ Р 34.10-94 и ГОСТ Р 34.10-2001 с синтаксисом криптографических сообщений (CMS)» (Leontiev, S., Ed. and G. Chudov, Ed., 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)), RFC 4490, май 2006.

[**IETF RFC 4648**] С. Юсефссон «Кодировки Base16, Base32 и Base64» (S. Josefsson, The Base16, Base32, and Base64 Data Encodings), RFC 4648, октябрь 2006.

УДК 351.864.1:004:083.131 **ОКС 35.040** **П85**

Ключевые слова:, *электронная коммерция, электронная цифровая подпись, алгоритмы шифрования, безопасность*

Руководитель организации-разработчика:

Генеральный директор
ООО «КРИПТО-ПРО»

_____ Чернова Н.Г.

Руководитель разработки:

Директор по науке
ООО «КРИПТО-ПРО»

_____ Попов В.О.

Авторы документа:

Технический Директор
ООО «КРИПТО-ПРО»

_____ Леонтьев С.Е.

ООО «Крипто-Про»

_____ Непомнящий П.