
ФЕДЕРАЛЬНОЕ АГЕНТСТВО ПО ТЕХНИЧЕСКОМУ РЕГУЛИРОВАНИЮ И МЕТРОЛОГИИ
(РОССТАНДАРТ)

Технический комитет 026

«Криптографическая защита информации»

Информационная технология

КРИПТОГРАФИЧЕСКАЯ ЗАЩИТА ИНФОРМАЦИИ

ТЕХНИЧЕСКАЯ СПЕЦИФИКАЦИЯ

**ИСПОЛЬЗОВАНИЯ АЛГОРИТМОВ ГОСТ Р 34.10, ГОСТ Р 34.11
В ПРОФИЛЕ СЕРТИФИКАТА И СПИСКЕ ОТЗЫВА СЕРТИФИКАТОВ (CRL)
ИНФРАСТРУКТУРЫ ОТКРЫТЫХ КЛЮЧЕЙ
X.509**

*Утверждена решением заседания
технического комитета по стандартизации
«Криптографическая защита информации»
(Протокол №13 от 24.04.2014 г.)*

Москва
2014

Содержание

Введение	3
1 Область применения	4
2 Нормативные ссылки.....	4
2.1 Дополнительные ссылки.....	5
3 Определения и обозначения.....	5
3.1 Определения.....	5
4 Поддерживаемые алгоритмы.....	6
4.1 Функция хэширования.....	6
4.1.1 Функции хэширования ГОСТ Р 34.11	6
4.2 Алгоритмы подписи.....	6
4.2.1 Алгоритмы подписи согласно ГОСТ Р 34.10.....	6
4.3 Алгоритмы открытого ключа субъекта.....	7
4.3.1 Открытые ключи согласно ГОСТ Р 34.10	7
5 Вопросы безопасности	9
6 Требования по совместимости.....	9
Приложение А :Параметры ГОСТ Р 34.10-2012 длины 256 бит (нормативное).	10
Приложение Б :Примеры (информативное)	13
7 Библиография.....	20

Введение

Настоящая рекомендация содержит описание форматов кодирования, идентификаторов и форматов параметров для алгоритмов по ГОСТ Р 34.10 и ГОСТ Р 34.11 при их использовании в инфраструктуре открытых ключей (PKI) X.509 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, а также алгоритмов согласования ключей VKO GOST R 34.10-2012 в инфраструктуре открытых ключей (PKI) российскими производителями

1 Область применения

Настоящая рекомендация является дополнением к международному стандарту IETF RFC 3279 и к государственному стандарту ГОСТ Р ИСО/МЭК 9594-8. В документе описываются правила использования алгоритма подписи ГОСТ Р 34.10, функции хэширования по ГОСТ Р 34.11, а также алгоритма согласования ключей VKO GOST R 34.10-2012, в инфраструктуре открытых ключей (PKI) X.509 Интернет [IETF RFC 5280] для вновь разрабатываемых систем PKI.

Для открытых ключей субъектов, использующих алгоритмы по ГОСТ Р 34.10 / VKO GOST R 34.10-2012 [TK26АЛГ], определены идентификаторы алгоритмов и соответствующие этим алгоритмам параметры. Также в документе указаны идентификаторы алгоритмов функции хэширования по ГОСТ Р 34.11 с алгоритмом подписи по ГОСТ Р 34.10.

В настоящем документе указан формат кодирования электронной подписи, сформированной с помощью алгоритмов ГОСТ Р 34.10.

В настоящем документе определено содержимое полей `signature`, `signatureAlgorithm`, `signatureValue` и `subjectPublicKey` в сертификатах X.509 и списках отзыва сертификатов. Для каждого алгоритма цифровой подписи предоставляется перечень возможных значений расширения `keyUsage` сертификата ключа подписи.

Абстрактная синтаксическая нотация версии один определена в ГОСТ Р ИСО/МЭК 8824-1. Дополнительные определения АСН.1, использованные в настоящем документе, можно найти в [TK26ЭК], [TK26УЗ] и [TK26АЛГ].

2 Нормативные ссылки

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.

ГОСТ Р ИСО/МЭК 9594-8 — «Информационные технологии. Взаимосвязь открытых систем. Справочник. Часть 8. Основы аутентификации», ГОСТ Р ИСО/МЭК 9594-8-98, Государственный стандарт Российской Федерации, Госстандарт России, Москва, 1998.

TK26АЛГ — Федеральное агентство по техническому регулированию и метрологии (РОССТАНДАРТ), Технический комитет №26, «Рекомендации по стандартизации. Использование криптографических алгоритмов, сопутствующих применению стандартов ГОСТ Р 34.10-2012 и ГОСТ Р 34.11-2012», Москва, 2014.

TK26ЭК — Федеральное агентство по техническому регулированию и метрологии (РОССТАНДАРТ), Технический комитет №26, «Рекомендации по стандартизации. Задание параметров эллиптических кривых в соответствии с ГОСТ Р 34.10-2012», Москва, 2013.

TK26УЗ — Федеральное агентство по техническому регулированию и метрологии (РОССТАНДАРТ), Технический комитет №26, «Рекомендации по стандартизации. Задание узлов замены блока подстановки алгоритма шифрования ГОСТ 28147-89», Москва, 2013.

2.1 Дополнительные ссылки

IETF RFC 3279 — Басгам, Л., Полк, У., и Р. Хаусли, «Алгоритмы и идентификаторы профиля сертификата и списка отзыва сертификатов инфраструктуры открытых ключей Интернет X.509» (Bassham, L., Polk, W., and R. Housley, "Algorithms and Identifiers for the Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile"), RFC 3279, апрель 2002.

IETF RFC 5280 — Д. Купер, С. Сэнтессон, С. Фаррел, С. Бойан, Р. Хаусли и У. Полк, «Профиль сертификата и списка отзыва сертификатов инфраструктуры открытых ключей Интернет X.509» (Cooper, D., Santesson, S., Farrell, S., Boeyen, S., Housley, R., and W. Polk, "Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile"), RFC 5280, май 2008.

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 и обозначения

3.1 Определения

В настоящем документе определены следующие термины:

закрытый ключ: Элемент секретных данных, специфичный для субъекта и используемый только данным субъектом в процессах согласования ключей, расшифрования ключей и формирования цифровой подписи (ключ подписи [ГОСТ Р 34.10])

открытый ключ: Элемент данных, математически связанный с закрытым ключом и используемый передающей стороной в процессах согласования ключей, шифрования ключей, а так же проверяющей стороной в процессе проверки цифровой подписи (ключ проверки подписи [ГОСТ Р 34.10], ключ общего пользования [ГОСТ Р ИСО/МЭК 9594-8])

[электронная цифровая] подпись (signature): Строка бит, полученная в результате процесса формирования подписи. Данная строка имеет внутреннюю структуру, зависящую от конкретного механизма формирования подписи. [ГОСТ Р 34.10]

Примечание – В настоящем документе в целях сохранения терминологической преемственности с действующими отечественными нормативными документами и опубликованными научно-техническими изданиями установлено, что термины «цифровая подпись», «электронная подпись» и «электронная цифровая подпись (ЭЦП)» являются синонимами.

4 Поддерживаемые алгоритмы

Данный раздел содержит описание криптографических алгоритмов, которые могут использоваться в профиле сертификата и списка отзыва сертификатов инфраструктуры открытых ключей Интернет X.509 [IETF RFC 5280] и [ГОСТ Р ИСО/МЭК 9594-8]. В настоящем разделе описываются функция хэширования и алгоритмы цифровой подписи, которые могут использоваться для формирования подписи сертификатов и списка отзыва сертификатов. Здесь также определены идентификаторы объектов (OID) и правила кодирования АСН.1 для содержащихся в сертификате открытых ключей.

Соответствующие данной рекомендации удостоверяющие центры и (или) приложения ДОЛЖНЫ поддерживать как минимум один из указанных алгоритмов открытых ключей и подписи.

4.1 Функция хэширования

В данном разделе описывается использование алгоритмов хэширования ГОСТ Р 34.11 которые можно использовать в алгоритме цифровой подписи ГОСТ Р 34.10. При этом ГОСТ Р 34.11 допустимо использовать совместно с ГОСТ Р 34.10 при условии соответствия размера хэш-кода и размера ключа подписи.

Полный перечень данных, хэшируемых для формирования подписи сертификатов и списков отзыва сертификатов, приведён в IETF RFC 5280, ГОСТ Р ИСО/МЭК 9594-8.

4.1.1 Функции хэширования ГОСТ Р 34.11

Алгоритмы по ГОСТ Р 34.11 используются для вычисления либо 256-битного, либо 512-битного хэш-кода исходных данных произвольной длины.

4.2 Алгоритмы подписи

Соответствующие настоящей рекомендации удостоверяющие центры могут использовать алгоритмы подписи по ГОСТ Р 34.10 для формирования подписи сертификатов и списков отзыва сертификатов.

Данные алгоритмы подписи ДОЛЖНЫ всегда использоваться с функциями хэширования по ГОСТ Р 34.11 в порядке, указанном в ГОСТ Р 34.10 и п. 4.1.данного документа.

В данном разделе определены идентификаторы и параметры алгоритмов для использования в поле signatureAlgorithm сертификата (Certificate) или списка сертификатов (CertificateList).

4.2.1 Алгоритмы подписи согласно ГОСТ Р 34.10

Идентификатор объекта АСН.1, используемый для определения алгоритма подписи на основе ГОСТ Р 34.11-2012 и ГОСТ Р 34.10-2012 со значением длины хэш-кода 256 бит:

```
id-tc26-signwithdigest-gost3410-2012-256 OBJECT IDENTIFIER ::=
  { iso(1) member-body(2) ru(643) rosstandart(7) tc26(1) algorithms(1)
    signwithdigest(3) gost3410-2012-256(2) }
```

Идентификатор объекта АСН.1, используемый для определения алгоритма подписи на основе ГОСТ Р 34.11-2012 и ГОСТ Р 34.10-2012 со значением длины хэш-кода 512 бит:

```
id-tc26-signwithdigest-gost3410-2012-512 OBJECT IDENTIFIER ::=
  { iso(1) member-body(2) ru(643) rosstandart(7) tc26(1) algorithms(1)
    signwithdigest(3) gost3410-2012-512(3) }
```

При кодировании НУЖНО опускать поле параметры (parameters). Таким образом, идентификатор AlgorithmIdentifier ДОЛЖЕН являться последовательностью (SEQUENCE), состоящей из одного компонента: идентификатора объекта (OBJECT IDENTIFIER) соответствующего алгоритма подписи.

Алгоритм подписи ГОСТ Р 34.10-2012 с длиной хэш-кода 256 бит используется для формирования цифровой подписи в форме двух 256-битных чисел, r и s . Её представление в виде строки октетов (OCTET STRING) идентично представлению подписи ГОСТ Р 34.10-2001 [IETF RFC 4491] 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.

Алгоритм подписи ГОСТ Р 34.10-2012 с длиной хэш-кода 512 используется для формирования цифровой подписи в форме двух 512-битных чисел, открытые ключи согласно r и s . 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.

Для преобразования данного представления в виде строки октетов в битовую строку, при использовании в сертификатах и списках отзыва сертификатов, ДОЛЖЕН использоваться процесс, описанный в разделе 4.3.1 настоящего документа.

4.3 Алгоритмы открытого ключа субъекта

В данном разделе определены идентификаторы объектов (OID) и параметры открытого ключа.

4.3.1 Открытые ключи согласно ГОСТ Р 34.10

Открытые ключи по ГОСТ Р 34.10 можно использовать для алгоритма подписи по ГОСТ Р 34.10, а также для алгоритма согласования ключей VKO GOST R 34.10-2012 [TK26ALG].

Открытые ключи по ГОСТ Р 34.10-2012 с ключом 256 бит определяются следующими идентификаторами объекта:

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

Открытые ключи по ГОСТ Р 34.10-2012 с ключом 512 бит определяются следующими идентификаторами объекта:

```
id-tc26-gost3410-2012-512 OBJECT IDENTIFIER ::=
    { iso(1) member-body(2) ru(643) rosstandart(7) tc26(1) algorithms(1)
      sign(1) gost3410-2012-512(2) }
```

В поле алгоритм (SubjectPublicKeyInfo.algorithm.algorithm) (см. IETF RFC 5280) для ключей по ГОСТ Р 34.10-2012 ДОЛЖНО быть указано соответствующее значение id-tc26-gost3410-2012-256 или id-tc26-gost3410-2012-512.

При кодировании МОЖНО опускать поле параметры (parameters) или устанавливать его значение в NULL. Для открытых ключей на которых разрешено передавать симметричные ключи СЛЕДУЕТ указывать поле encryptionParamSet. Параметры открытого ключа ДОЛЖНЫ иметь следующую структуру:

```
GostR3410-2012-PublicKeyParameters ::=
    SEQUENCE {
        publicKeyParamSet          OBJECT IDENTIFIER,
        digestParamSet             OBJECT IDENTIFIER,
        encryptionParamSet         OBJECT IDENTIFIER DEFAULT
```

```
        id-tc26-gost-28147-param-z
    }
```

где:

- `publicKeyParamSet` – идентификатор параметров открытого ключа по ГОСТ Р 34.10;
- `digestParamSet` – идентификатор алгоритма и параметров по ГОСТ Р 34.11;
- `encryptionParamSet` – идентификатор алгоритма и параметров по ГОСТ 28147.

Отсутствие параметров следует обрабатывать в порядке, описываемом в **IETF RFC 5280**, раздел 6.1, то есть параметры должны быть унаследованы из сертификата издателя. Если переменная `working_public_key_parameters` установлена в нулевое значение, то сам сертификат и любая проверяемая подпись, созданная с использованием закрытого ключа, соответствующего данному сертификату, ДОЛЖНЫ быть отклонены.

Согласно стандарту ГОСТ Р 34.10 открытый ключ является точкой на эллиптической кривой:

$$Q = (x, y)$$

Представление открытого ключа `GostR3410-2012-256-PublicKey` идентично представлению открытого ключа ГОСТ Р 34.10-2001 **[IETF RFC 4491]**, и ДОЛЖНО содержать 64 октета, где первые 32 октета содержат координату x в представлении `little-endian`, и вторые 32 октета содержат координату y в представлении `little-endian`.

Представление открытого ключа `GostR3410-2012-512-PublicKey` ДОЛЖНО содержать 128 октетов, где первые 64 октета содержат координату x в представлении `little-endian`, и вторые 64 октета содержат координату y в представлении `little-endian`.

Открытый ключ по ГОСТ Р 34.10 ДОЛЖЕН быть закодирован в DER как строка октетов (OCTET STRING) в соответствии с ГОСТ Р ИСО/МЭК 8825-1-2003:

```
GostR3410-2012-256-PublicKey ::= OCTET STRING -- вектор открытого ключа, Q
GostR3410-2012-512-PublicKey ::= OCTET STRING -- вектор открытого ключа, Q
```

Далее, результат этого кодирования используется в качестве значения компонента `subjectPublicKey` структуры `SubjectPublicKeyInfo`. Поскольку значение `subjectPublicKey` должно быть представлено в виде битовой строки (BIT STRING), необходимо преобразование значения цифровой подписи из строки октетов (OCTET STRING) в битовую строку (BIT STRING).

Для преобразования значения подписи из строки октетов (OCTET STRING) в битовую строку (BIT STRING) наименее значащий бит первого октета строки октетов (OCTET STRING) становится младшим битом битовой строки и так далее вплоть до наиболее значащего бита последнего октета OCTET STRING, который становится последним битом битовой строки.

Если в сертификате конечного пользователя, содержащем открытый ключ по ГОСТ Р 34.10, присутствует расширение `keyUsage`, в нем МОГУТ присутствовать следующие значения:

- `digitalSignature`;
- `nonRepudiation`;
- `keyEncipherment`;
- `keyAgreement`.

Если в сертификате удостоверяющего центра или сертификате ключа подписи списка отзыва сертификатов, содержащем открытый ключ по ГОСТ Р 34.10, присутствует расширение `keyUsage`, в нем МОГУТ присутствовать следующие значения:

- `digitalSignature`;

- nonRepudiation;
- keyCertSign;
- cRLSign.

5 Вопросы безопасности

РЕКОМЕНДУЕТСЯ, чтобы приложения проверяли значения подписи и открытые ключи на предмет их соответствия стандарту ГОСТ Р 34.10 до начала их использования.

РЕКОМЕНДУЕТСЯ, чтобы удостоверяющие центры и приложения следили за тем, чтобы закрытый ключ электронной подписи не использовался дольше допустимого срока действия (как правило, 15 месяцев для ключей алгоритма ГОСТ Р 34.10).

6 Требования по совместимости

Требования по реализации X.509 на основе 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 [ТК26ЭК];
- id-tc26-gost-3410-12-512-paramSetA — при поддержке 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 шифрования ключей (keyEncipherment) [ТК26УЗ].

Приложение А : Параметры ГОСТ Р 34.10-2012 длины 256 бит (нормативное).

Параметры открытых ключей ГОСТ Р 34.10-2012 длины 256 бит идентичны параметрам id-GostR3410-2001-CryptoPro-XchA-ParamSet, id-GostR3410-2001-CryptoPro-XchB-ParamSet, id-GostR3410-2001-CryptoPro-A-ParamSet, id-GostR3410-2001-CryptoPro-B-ParamSet и id-GostR3410-2001-CryptoPro-C-ParamSet, открытых ключей ГОСТ Р 34.10-2001 [IETF RFC 4357].

Идентификаторы объектов (OID) АСН.1:

```
id-GostR3410-2001-CryptoPro-XchA-ParamSet OBJECT IDENTIFIER ::=
  { iso(1) member-body(2) ru(643) rans(2) cryptopro(2)
    ecc-exchanges(36) cryptopro-XchA(0) }
id-GostR3410-2001-CryptoPro-XchB-ParamSet OBJECT IDENTIFIER ::=
  { iso(1) member-body(2) ru(643) rans(2) cryptopro(2)
    ecc-exchanges(36) cryptopro-XchB(1) }
id-GostR3410-2001-CryptoPro-A-ParamSet OBJECT IDENTIFIER ::=
  { iso(1) member-body(2) ru(643) rans(2) cryptopro(2)
    ecc-signs(35) cryptopro-A(1) }
id-GostR3410-2001-CryptoPro-B-ParamSet OBJECT IDENTIFIER ::=
  { iso(1) member-body(2) ru(643) rans(2) cryptopro(2)
    ecc-signs(35) cryptopro-B(2) }
id-GostR3410-2001-CryptoPro-C-ParamSet OBJECT IDENTIFIER ::=
  { iso(1) member-body(2) ru(643) rans(2) cryptopro(2)
    ecc-signs(35) cryptopro-C(3) }
```

Значения параметров:

```
678 30 159: SEQUENCE {
681 06 7: OBJECT IDENTIFIER
      : id-GostR3410-2001-CryptoPro-XchA-ParamSet
690 30 147: SEQUENCE {
693 02 33: INTEGER
      : 00 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
      : FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FD
      : 94
728 02 2: INTEGER 166
732 02 33: INTEGER
      : 00 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
      : FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FD
      : 97
767 02 33: INTEGER
      : 00 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
      : FF 6C 61 10 70 99 5A D1 00 45 84 1B 09 B7 61 B8
      : 93
802 02 1: INTEGER 1
805 02 33: INTEGER
      : 00 8D 91 E4 71 E0 98 9C DA 27 DF 50 5A 45 3F 2B
      : 76 35 29 4F 2D DF 23 E3 B1 22 AC C9 9C 9E 9F 1E
      : 14
      : }
      : }
840 30 159: SEQUENCE {
843 06 7: OBJECT IDENTIFIER
      : id-GostR3410-2001-CryptoPro-XchB-ParamSet
852 30 147: SEQUENCE {
855 02 33: INTEGER
      : 00 9B 9F 60 5F 5A 85 81 07 AB 1E C8 5E 6B 41 C8
```

```

      :   AA CF 84 6E 86 78 90 51 D3 79 98 F7 B9 02 2D 75
      :   98
890 02  3:   INTEGER 32858
895 02 33:   INTEGER
      :   00 9B 9F 60 5F 5A 85 81 07 AB 1E C8 5E 6B 41 C8
      :   AA CF 84 6E 86 78 90 51 D3 79 98 F7 B9 02 2D 75
      :   9B
930 02 33:   INTEGER
      :   00 9B 9F 60 5F 5A 85 81 07 AB 1E C8 5E 6B 41 C8
      :   AA 58 2C A3 51 1E DD FB 74 F0 2F 3A 65 98 98 0B
      :   B9
965 02  1:   INTEGER 0
968 02 32:   INTEGER
      :   41 EC E5 57 43 71 1A 8C 3C BF 37 83 CD 08 C0 EE
      :   4D 4D C4 40 D4 64 1A 8F 36 6E 55 0D FD B3 BB 67
      :   }
      :   }
163 30 159:  SEQUENCE {
166 06  7:   OBJECT IDENTIFIER
      :   id-GostR3410-2001-CryptoPro-A-ParamSet
175 30 147:  SEQUENCE {
178 02 33:   INTEGER
      :   00 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
      :   FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FD
      :   94
213 02  2:   INTEGER 166
217 02 33:   INTEGER
      :   00 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
      :   FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FD
      :   97
252 02 33:   INTEGER
      :   00 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
      :   FF 6C 61 10 70 99 5A D1 00 45 84 1B 09 B7 61 B8
      :   93
287 02  1:   INTEGER 1
290 02 33:   INTEGER
      :   00 8D 91 E4 71 E0 98 9C DA 27 DF 50 5A 45 3F 2B
      :   76 35 29 4F 2D DF 23 E3 B1 22 AC C9 9C 9E 9F 1E
      :   14
      :   }
      :   }
325 30 188:  SEQUENCE {
328 06  7:   OBJECT IDENTIFIER
      :   id-GostR3410-2001-CryptoPro-B-ParamSet
337 30 176:  SEQUENCE {
340 02 33:   INTEGER
      :   00 80 00 00 00 00 00 00 00 00 00 00 00 00 00 00
      :   00 00 00 00 00 00 00 00 00 00 00 00 00 00 0C
      :   96
375 02 32:   INTEGER
      :   3E 1A F4 19 A2 69 A5 F8 66 A7 D3 C2 5C 3D F8 0A
      :   E9 79 25 93 73 FF 2B 18 2F 49 D4 CE 7E 1B BC 8B
409 02 33:   INTEGER
      :   00 80 00 00 00 00 00 00 00 00 00 00 00 00 00 00
      :   00 00 00 00 00 00 00 00 00 00 00 00 00 00 0C
      :   99
444 02 33:   INTEGER

```

```

      :      00 80 00 00 00 00 00 00 00 00 00 00 00 00 00 00
      :      01 5F 70 0C FF F1 A6 24 E5 E4 97 16 1B CC 8A 19
      :      8F
479 02   1:    INTEGER 1
482 02   32:   INTEGER
      :      3F A8 12 43 59 F9 66 80 B8 3D 1C 3E B2 C0 70 E5
      :      C5 45 C9 85 8D 03 EC FB 74 4B F8 D7 17 71 7E FC
      :      }
      :    }
516 30  159:  SEQUENCE {
519 06   7:    OBJECT IDENTIFIER
      :      id-GostR3410-2001-CryptoPro-C-ParamSet
528 30  147:  SEQUENCE {
531 02   33:   INTEGER
      :      00 9B 9F 60 5F 5A 85 81 07 AB 1E C8 5E 6B 41 C8
      :      AA CF 84 6E 86 78 90 51 D3 79 98 F7 B9 02 2D 75
      :      98
566 02   3:    INTEGER 32858
571 02   33:   INTEGER
      :      00 9B 9F 60 5F 5A 85 81 07 AB 1E C8 5E 6B 41 C8
      :      AA CF 84 6E 86 78 90 51 D3 79 98 F7 B9 02 2D 75
      :      9B
606 02   33:   INTEGER
      :      00 9B 9F 60 5F 5A 85 81 07 AB 1E C8 5E 6B 41 C8
      :      AA 58 2C A3 51 1E DD FB 74 F0 2F 3A 65 98 98 0B
      :      B9
641 02   1:    INTEGER 0
644 02   32:   INTEGER
      :      41 EC E5 57 43 71 1A 8C 3C BF 37 83 CD 08 C0 EE
      :      4D 4D C4 40 D4 64 1A 8F 36 6E 55 0D FD B3 BB 67
      :      }
      :    }

```

Приложение Б : Примеры (информативное)

Сертификат в кодировке Base64 [IETF RFC 4648]:

```
-----BEGIN CERTIFICATE-----
MIICYjCCAg+gAwIBAgIBATAKBggqhQMHAQEDAJBWMSkwJwYJKoZIhvcNAQkBFhpH
b3N0UjM0MTAtMjAxMkBlEgFtcGx1LmNvbTEpMCCGAlUEAxMGR29zdFIZNDEwLTIw
MTIqKDI1NiBiaXQpIGV4YW1wbGUwHhcNMTMxMTA1MTQwMjM3WhcNMzAxMTAxMTQw
MjM3WjBWMSkwJwYJKoZIhvcNAQkBFhpHb3N0UjM0MTAtMjAxMkBlEgFtcGx1LmNv
bTEpMCCGAlUEAxMGR29zdFIZNDEwLTIwMTIqKDI1NiBiaXQpIGV4YW1wbGUwZjAf
BggqhQMHAQEBATATBgcqhQMCAiQABggqhQMHAQECAGNDAARAut/Qw1MUq9KPqkdH
C2xAF3K7TugHfo9n525D2s5mFZdD5pwf90/i4vF0mFmr9nfrWmYP4o0Pg1mOn5Rl
aXNYraOBwDCBvTAdBgNVHQ4EFgQU1fIeN1HaPbw+XWUzbkJ+kHJUT0AwCwYDVR0P
BAQDAgHGMA8GA1UdEwQIMAYBAf8CAQEwfGyDVR0BBHcWdYAU1fIeN1HaPbw+XWUz
bkJ+kHJUT0ChWqRYMFYxKTANBgkqhkiG9w0BCQEWGkdvc3RSMzQxMC0yMDEyQGV4
YW1wbGUuY29tMSkwJwYDVQQDEYBhb3N0UjM0MTAtMjAxMjAxMjU2IGJpdCkgZXhh
bXBsZS9YIBATAKBggqhQMHAQEDAGNBAF5bm4BbARR6hJLEoWJkOsYV3Hd7kXQQjz3C
dqQfmHrz6TI6Xojdh/t8ckODv/587NS5/6KsM77vc6Wh90NAT2s=
-----END CERTIFICATE-----
```

ACH.1 представление сертификата:

```
0000 30 02 62: SEQUENCE {
0004 30 02 0f: SEQUENCE {
0008 a0 03: [0] {
000a 02 01: INTEGER 02
: }
000d 02 01: INTEGER 01
0010 30 0a: SEQUENCE {
0012 06 08: OBJECT IDENTIFIER
0014 : id-tc26-signwithdigest-gost3410-2012-256
: (1 2 643 7 1 1 3 2)
: }
001c 30 56: SEQUENCE {
001e 31 29: SET {
0020 30 27: SEQUENCE {
0022 06 09: OBJECT IDENTIFIER emailAddress (1 2 840 113549 1 9 1)
002d 16 1a: IA5 STRING 'GostR3410-2012@example.com'
: }
: }
0049 31 29: SET {
004b 30 27: SEQUENCE {
004d 06 03: OBJECT IDENTIFIER commonName (2 5 4 3)
0052 13 20: PRINTABLE STRING 'GostR3410-2012 (256 bit) example'
: }
: }
: }
0074 30 1e: SEQUENCE {
0076 17 0d: UTCTime '131105140237Z'
0085 17 0d: UTCTime '301101140237Z'
: }
0094 30 56: SEQUENCE {
0096 31 29: SET {
0098 30 27: SEQUENCE {
009a 06 09: OBJECT IDENTIFIER emailAddress (1 2 840 113549 1 9 1)
00a5 16 1a: IA5 STRING 'GostR3410-2012@example.com'
```

```

      :      }
      :      }
00c1 31    29:    SET {
00c3 30    27:      SEQUENCE {
00c5 06    03:        OBJECT IDENTIFIER commonName (2 5 4 3)
00ca 13    20:        PRINTABLE STRING 'GostR3410-2012 (256 bit) example'
      :      }
      :      }
      :    }
00ec 30    66:    SEQUENCE {
00ee 30    1f:      SEQUENCE {
00f0 06    08:        OBJECT IDENTIFIER
      :          id-tc26-gost3410-2012-256
      :          (1 2 643 7 1 1 1 1)
00fa 30    13:      SEQUENCE {
00fc 06    07:        OBJECT IDENTIFIER
      :          id-GostR3410-2001-CryptoPro-XchA-ParamSet
      :          (1 2 643 2 2 36 0)
0105 06    08:        OBJECT IDENTIFIER
      :          id-tc26-gost3411-2012-256
      :          (1 2 643 7 1 1 2 2)
      :      }
      :    }
010f 03    43:    BIT_STRING 0 unused bits, encapsulates {
0112 04    40:      OCTET STRING
      :        ba df d0 c3 53 14 ab d2 8f aa 47 47 0b 6c 40 17
      :        72 bb 4e e8 07 7e 8f 67 e7 6e 43 da ce 66 15 97
      :        43 e6 9c 1f f7 4f e2 e2 f1 74 98 59 ab f6 77 d1
      :        c0 c6 0f e2 8d 0f 83 59 8e 9f 94 65 69 73 58 ad
      :      }
      :    }
0154 a3    c0:    [3] {
0157 30    bd:      SEQUENCE {
015a 30    1d:        SEQUENCE {
015c 06    03:          OBJECT IDENTIFIER
      :            subjectKeyIdentifier (2 5 29 14)
0161 04    16:        OCTET STRING, encapsulates {
0163 04    14:          OCTET_STRING
      :            d5 f2 1e 37 51 da 3d bc 3e 5d 65 33 6e 42 7e 90
      :            72 54 4f 40
      :          }
      :        }
0179 30    0b:      SEQUENCE {
017b 06    03:        OBJECT IDENTIFIER
      :          keyUsage (2 5 29 15)
0180 04    04:        OCTET STRING, encapsulates {
0182 03    02:          BIT_STRING
      :            01 c6
      :          }
      :        }
0186 30    0f:      SEQUENCE {
0188 06    03:        OBJECT IDENTIFIER
      :          basicConstraints (2 5 29 19)
018d 04    08:        OCTET STRING, encapsulates {
018f 30    06:          SEQUENCE {
0191 01    01:            BOOL ff
0194 02    01:            INTEGER 01

```


Соответствующий закрытый ключ d равен:

0xBFCF1D623E5CDD3032A7C6EABV4A923C46E43D640FFEAAF2C3ED39A8FA399924

Число \bar{h} равно:

0x706FA77A1F5ECDFA171B7ACB2128A0E6A4D26F3C0FFB2EF283B16CEA207E061C

Число k равно:

0x5782C53F110C596F9155D35EBD25A06A89C50391850A8FEFE33B0E270318857C

В подписи сертификата, r равно:

0xE9323A5E88DD87FB7C724383BFFE7CECD4B9FFA2AC33BEEF73A5A1F743404F6B

s равно:

0x5E5B9B805B01147A8492C4A162643AC615DC777B9174108F3DC276A41F987AF3

```
-----BEGIN CERTIFICATE-----
MIIC6DCCA1SgAwIBAgIBATAKBggqhQMHAQEDAzBWMSkwJwYJKoZIhvcNAQkBFhpH
b3N0UjM0MTAtMjAxMkBlcGFtcGxlLmNvbTEpMCCGAlUEAxMGR29zdFIZNDEwLTlw
MTIwIGKDUxMiBiaXQpIGV4YW1wbGUwHhcNMTMxMDA0MDczNjA0WhcNMzAxMDAxMDcz
NjA0WjBWMScwJwYJKoZIhvcNAQkBFhpHb3N0UjM0MTAtMjAxMkBlcGFtcGxlLmNv
bTEpMCCGAlUEAxMGR29zdFIZNDEwLTlwMTIwIGKDUxMiBiaXQpIGV4YW1wbGUwgaow
IQYIKoUDBwEBAQIwFQYJKoUDBwECAQICBggqhQMHAQECAwOBhAAEgYATGQ9VCiM5
FRGCQ8MEz2F1dANqhaEuywa8CbxOnTvaGJpFQVXQwkwvLFAKh7hk542vOEtXPkTt
CXfGf84nRhMH/Q9bZeAc2eO/yhXrsQhTBufa1Fuou2oe/jUOaG6RatUUvRzhNTpp
RGG11+EIY2vzzUua9j90l/gAoy/LNKQIfqOBwDCBvTAdBgNVHQ4EFgQUPcbTRXJZ
nHtjj+eBP7b51cTMekIwCwYDVR0PBAQDAgHGMA8GA1UdEwQIMAYBAf8CAQEwfgYD
VR0BBHcWdYAUPcbTRXJZnHtjj+eBP7b51cTMekKhWqRYMFYxKTAnBgkqhkiG9w0B
CQEwGkdvc3RSMzQxMCM0YmDEyQGV4YW1wbGUuY29tMSkwJwYDVRQDEyBHb3N0UjM0
MTAtMjAxMjA0NTEyIGJpdCkgZXhhbXBsZS9IBATAKBggqhQMHAQEDAwOBgQBObs7o
ppPTXzHyVR1DtPa8b57nudJzI4czhsfeX5HDntOq45t9B/qSs8dC6eGxbhHZ9zCO
SFtxWYdmg0au8XI9Xb8vTC1qdwWID7FFjMWDNQZb61Yh/J+8F2xKy1vB5nI1RZqO
o3eUNFkNyHJwQCk2Wo0l016zwGk2tdKH4KmD5w==
-----END CERTIFICATE-----
```

ASN.1 представление сертификата:

```
0000 30 02 e8: SEQUENCE {
0004 30 02 54: SEQUENCE {
0008 a0 03: [0] {
000a 02 01: INTEGER 02
: }
000d 02 01: INTEGER 01
0010 30 0a: SEQUENCE {
0012 06 08: OBJECT IDENTIFIER
0014 : id-tc26-signwithdigest-gost3410-2012-512
: (1 2 643 7 1 1 3 3)
: }
001c 30 56: SEQUENCE {
001e 31 29: SET {
0020 30 27: SEQUENCE {
0022 06 09: OBJECT IDENTIFIER emailAddress (1 2 840 113549 1 9 1)
002d 16 1a: IA5 STRING 'GostR3410-2012@example.com'
: }
: }
0049 31 29: SET {
004b 30 27: SEQUENCE {
```



```

004d 06      03:      OBJECT IDENTIFIER commonName (2 5 4 3)
0052 13      20:      PRINTABLE STRING 'GostR3410-2012 (512 bit) example'
           :      }
           :      }
           :      }
0074 30      1e:      SEQUENCE {
0076 17      0d:      UTCTime '131004073604Z'
0085 17      0d:      UTCTime '301001073604Z'
           :      }
0094 30      56:      SEQUENCE {
0096 31      29:      SET {
0098 30      27:      SEQUENCE {
009a 06      09:      OBJECT IDENTIFIER emailAddress (1 2 840 113549 1 9 1)
00a5 16      1a:      IA5 STRING 'GostR3410-2012@example.com'
           :      }
           :      }
00c1 31      29:      SET {
00c3 30      27:      SEQUENCE {
00c5 06      03:      OBJECT IDENTIFIER commonName (2 5 4 3)
00ca 13      20:      PRINTABLE STRING 'GostR3410-2012 (512 bit) example'
           :      }
           :      }
           :      }
00ec 30      aa:      SEQUENCE {
00ef 30      21:      SEQUENCE {
00f1 06      08:      OBJECT IDENTIFIER
           :      id-tc26-gost3410-2012-512
           :      (1 2 643 7 1 1 1 2)
00fb 30      15:      SEQUENCE {
00fd 06      09:      OBJECT IDENTIFIER
           :      id-tc26-gost-3410-2012-512-paramSetB
           :      (1 2 643 7 1 2 1 2 2)
0108 06      08:      OBJECT IDENTIFIER
           :      id-tc26-gost3411-2012-512
           :      (1 2 643 7 1 1 2 3)
           :      }
           :      }
0112 03      84:      BIT_STRING 0 unused bits, encapsulates {
0116 04      80:      OCTET STRING
           :      13 19 0f 55 0a 23 39 15 11 82 43 c3 04 cf 61 75
           :      74 03 6a 85 a1 2e cb 06 bc 09 bc 4e 9d 3b da 18
           :      9a 45 41 55 d0 c2 4c 2f 2c 50 0a 87 b8 64 e7 8d
           :      af 38 4b 71 a4 ab 53 09 77 c6 7f ce 27 46 13 07
           :      fd 0f 5b 65 e0 1c d9 e3 bf ca 1c 6b b1 08 53 06
           :      e7 da d4 5b a8 bb 6a 1e fe 35 0e 68 6e 91 02 d5
           :      14 bd 1c e1 35 3a 69 44 61 a5 d7 e1 08 63 6b f3
           :      cd 4b 9a f6 3f 4e 97 f8 00 a3 2f cb 34 a4 08 7e
           :      }
           :      }
0199 a3      c0:      [3] {
019c 30      bd:      SEQUENCE {
019f 30      1d:      SEQUENCE {
01a1 06      03:      OBJECT IDENTIFIER
           :      subjectKeyIdentifier (2 5 29 14)
01a6 04      16:      OCTET STRING, encapsulates {
01a8 04      14:      OCTET STRING
           :      3d c6 d3 45 72 59 9c 7b 63 8f e7 81 3f b6 f9 95

```

```

      :      c4 cc 7a 42
      :      }
      :      }
01be 30 0b: SEQUENCE {
01c0 06 03:   OBJECT IDENTIFIER
      :      keyUsage (2 5 29 15)
01c5 04 04:   OCTET STRING, encapsulates {
01c7 03 02:     BIT STRING
      :      01 c6
      :      }
      :      }
01cb 30 0f: SEQUENCE {
01cd 06 03:   OBJECT IDENTIFIER
      :      basicConstraints (2 5 29 19)
01d2 04 08:   OCTET STRING, encapsulates {
01d4 30 06:     SEQUENCE {
01d6 01 01:       BOOL ff
01d9 02 01:       INTEGER 01
      :       }
      :       }
      :       }
01dc 30 7e: SEQUENCE {
01de 06 03:   OBJECT IDENTIFIER
      :      authorityKeyIdentifier (2 5 29 1)
01e3 04 77:   OCTET STRING, encapsulates {
01e5 30 75:     SEQUENCE {
01e7 80 14:       [0]
      :       3d c6 d3 45 72 59 9c 7b 63 8f e7 81 3f b6 f9 95
      :       c4 cc 7a 42
01fd a1 5a:       [1] {
01ff a4 58:         [4] {
0201 30 56:           SEQUENCE {
0203 31 29:             SET {
0205 30 27:               SEQUENCE {
0207 06 09:                 OBJECT IDENTIFIER emailAddress (1 2 840 113549 1 9 1)
0212 16 1a:                 IA5 STRING 'GostR3410-2012@example.com'
      :                 }
      :               }
022e 31 29:             SET {
0230 30 27:               SEQUENCE {
0232 06 03:                 OBJECT IDENTIFIER commonName (2 5 4 3)
0237 13 20:                 PRINTABLE STRING 'GostR3410-2012 (512 bit) example'
      :                 }
      :               }
      :             }
      :           }
      :         }
      :       }
      :     }
      :   }
      : }
0259 82 01: [2]
      : 01
025c 30 0a: SEQUENCE {
025e 06 08:   OBJECT IDENTIFIER

```

```

      :      id-tc26-signwithdigest-gost3410-2012-512
      :      (1 2 643 7 1 1 3 3)
      :      }
0268 03 81: BIT STRING 0 unused bits
      :      4e 6d 2e e8 a6 93 d3 5f 31 f2 55 1d 43 b4 f6 bc
      :      6f 9e e7 b9 d2 73 23 87 33 86 c7 de 5f 91 c3 9e
      :      d3 aa e3 9b 7d 07 fa 92 b3 c7 42 e9 e1 b1 6e 11
      :      d9 f7 30 8e 48 5b 71 59 87 66 83 46 ae f1 72 3d
      :      5d bf 2f 4c 2d 6a 77 05 88 0f b1 45 8c c5 83 35
      :      06 5b ea 56 21 fc 9f bc 17 6c 4a ca 5b c1 e6 72
      :      25 45 9a 8e a3 77 94 34 59 0d c8 72 70 40 29 36
      :      5a 83 a5 3b 5e b3 c0 69 36 b5 d2 87 e0 a9 83 e7
      :      }

```

Координата x открытого ключа сертификата равна:

```

0x07134627CE7FC6770953ABA4714B38AF8DE764B8870A502C2F4CC2D05541459A18DA3B9D4EBC09BC06C
B2EA1856A03747561CF04C34382111539230A550F1913

```

Координата y равна:

```

0x7E08A434CB2FA300F8974E3FF69A4BCDF36B6308E1D7A56144693A35E11CBD14D502916E680E35FE1E6
ABBA85BD4DAE7065308B16B1CCABFE3D91CE0655B0FFD

```

Соответствующий закрытый ключ d равен:

```

0x3FC01CDD4EC5F972EB482774C41E66DB7F380528DFE9E67992BA05AEE462435757530E641077CE587B
976C8EEB48C48FD33FD175F0C7DE6A44E014E6BCB074B

```

Число \bar{h} равно:

```

0xC066476A9753A58A2EEE347FA7F7EC57FCA4C9D29B2172E23B988B7FA59D361D9AB25CAADB2C5338D98
966368441208F7A01195B7F7B45F1E4DD5FD4BE57C2ED

```

Число k равно:

```

0x72ABB44536656BF1618CE10BF7EADD40582304A51EE4E2A25A0A32CB0E773ABB23B7D8FDD8FA5EEE91B
4AE452F2272C86E1E2221215D405F51B5D5015616E1F6

```

В подписи сертификата, r равно:

```

0x5DBF2F4C2D6A7705880FB1458CC58335065BEA5621FC9FBC176C4ACA5BC1E67225459A8EA3779434590
DC872704029365A83A53B5EB3C06936B5D287E0A983E7

```

s равно:

```

0x4E6D2EE8A693D35F31F2551D43B4F6BC6F9EE7B9D27323873386C7DE5F91C39ED3AAE39B7D07FA92B3C
742E9E1B16E11D9F7308E485B715987668346AEF1723D

```

7 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 4491**] — Под ред. С. Леонтьева и Д. Шефановского «Использование алгоритмов по ГОСТ Р 34.10-94, ГОСТ Р 34.10-2001 и ГОСТ Р 34.11-94 в профиле сертификата и списка отзыва сертификатов (CLR) инфраструктуры открытых ключей Интернет X.509» (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, 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.