



**МИНИСТЕРСТВО ЦИФРОВОГО РАЗВИТИЯ, СВЯЗИ И МАССОВЫХ  
КОММУНИКАЦИЙ РОССИЙСКОЙ ФЕДЕРАЦИИ**

(МИНКОМСВЯЗЬ РОССИИ)

МИНИСТЕРСТВО ЮСТИЦИИ РОССИЙСКОЙ ФЕДЕРАЦИИ

**ЗАРЕГИСТРИРОВАНО**

Регистрационный № 60631

от "29 сентября 2020."

**ПРИКАЗ**

14.09.2020

№ 472

**Об утверждении Формата электронной подписи, обязательного для  
реализации всеми средствами электронной подписи**

В соответствии с положениями пункта 5 части 4 статьи 8 Федерального закона от 6 апреля 2011 г. № 63-ФЗ «Об электронной подписи» (Собрание законодательства Российской Федерации, 2011, № 15, ст. 2036; 2019, № 26, ст. 3889) и абзаца третьего пункта 1 Положения о Министерстве цифрового развития, связи и массовых коммуникаций Российской Федерации, утвержденного постановлением Правительства Российской Федерации от 2 июня 2008 г. № 418 (Собрание законодательства Российской Федерации, 2008, № 42, ст. 4825; 2018, № 40, ст. 6142),

**ПРИКАЗЫВАЮ:**

1. Утвердить Формат электронной подписи, обязательный для реализации всеми средствами электронной подписи, согласно приложению к настоящему приказу.
2. Направить настоящий приказ на государственную регистрацию в Министерство юстиции Российской Федерации.

Министр

М.И. Шадаев

**УТВЕРЖДЕН**  
приказом Министерства  
цифрового развития, связи  
и массовых коммуникаций  
Российской Федерации  
от 14.09.2020 № 472

## **ФОРМАТ**

### **электронной подписи, обязательный для реализации всеми средствами электронной подписи**

1. Настоящий формат электронной подписи, обязательный для реализации всеми средствами электронной подписи (далее – Формат), устанавливает требования к структуре и содержанию информации в электронной подписи (далее – ЭП).

В составе ЭП должна размещаться информация об исходном электронном сообщении, алгоритмах хэширования и ЭП, параметрах криптографических алгоритмов, времени создания ЭП, сертификат ключа проверки ЭП (далее – сертификат), иерархически обусловленная последовательность сертификатов, каждый последующий сертификат которой подписан ЭП, основанной на предшествующем сертификате, и иные, установленные в соответствии с пунктами 5, 6 и 7 настоящего Формата, сведения.

2. В соответствии с настоящим Форматом средства ЭП должны обеспечивать возможность создания нескольких ЭП в одном документе с сохранением данных, описывающих контекст, содержание, структуру документов, а также обеспечивающих управление документами в используемых информационных системах, – метаданных и сертификатов, на которых основаны эти ЭП, в электронном сообщении.

3. При включении в состав ЭП дополнительной информации, отличной от указанной в пунктах 5, 6 и 7 настоящего Формата, требования к ее назначению и расположению в структуре ЭП определяются заказчиком в техническом задании на разработку (модернизацию) средств ЭП и удостоверяющего центра (далее – УЦ), в соответствии с приказом ФСБ России от 9 февраля 2005 г. № 66 «Об утверждении Положения о разработке, производстве, реализации и эксплуатации шифровальных (криптографических) средств защиты информации (Положение ПКЗ-2005)» (зарегистрирован Минюстом России 3 марта 2005 г., регистрационный № 6382), с изменениями, внесенными приказом ФСБ России от 12 апреля 2010 г. № 173 «О внесении изменений в некоторые нормативные правовые акты ФСБ России» (зарегистрирован Минюстом России 25 мая 2010 г., регистрационный № 17350).

4. Формат определяется в соответствии с основами аутентификации

в открытых системах<sup>1</sup>, синтаксисом криптографических сообщений, описанием дополнительных атрибутов криптографических сообщений, спецификацией абстрактной синтаксической нотации версии один<sup>2</sup>.

5. В случае если участник электронного взаимодействия является владельцем действующего сертификата ключа проверки электронной подписи, Формат имеет вид следующей структуры:

```
SignedData ::= SEQUENCE {
    version                CMSVersion,
    digestAlgorithms       DigestAlgorithmIdentifiers,
    encapContentInfo       EncapsulatedContentInfo,
    certificates            [0] IMPLICIT CertificateSet OPTIONAL,
    crls                    [1] IMPLICIT RevocationInfoChoices OPTIONAL,
    signerInfos            SignerInfos }
```

5.1. Поле `version` (типа `CMSVersion`) определяет версию синтаксиса ЭП, которая зависит от сертификатов, типа подписываемых данных и информации о подписывающих сторонах:

```
CMSVersion ::= INTEGER
    { v0(0), v1(1), v2(2), v3(3), v4(4), v5(5) }
```

5.2. Поле `digestAlgorithms` (тип `DigestAlgorithmIdentifiers`) включает в себя идентификаторы используемых алгоритмов хэширования и связанные с ними параметры и определяется следующим образом:

```
DigestAlgorithmIdentifiers ::= SET OF DigestAlgorithmIdentifier
DigestAlgorithmIdentifier ::= AlgorithmIdentifier
```

В качестве идентификатора алгоритма хэширования указывается объектный идентификатор алгоритма, определенного ГОСТ Р 34.11-2012 «Информационная технология. Криптографическая защита информации. Функция хэширования»<sup>3</sup> (далее – ГОСТ Р 34.11-2012). Указанный объектный идентификатор в соответствии со спецификацией абстрактной синтаксической нотации версии один представляется следующим образом:

<sup>1</sup> Основы аутентификации в открытых системах определены в ГОСТ Р ИСО/МЭК 9594-8-98 «Информационная технология. Взаимосвязь открытых систем. Справочник. Часть 8. Основы аутентификации» (принят и введен в действие постановлением Госстандарта России от 19 мая 1998 г. № 215, опубликован в августе 1998 г. ИПК «Издательство стандартов», ИУС 9-98).

<sup>2</sup> Спецификация абстрактной синтаксической нотации версии один определена в ГОСТ Р ИСО/МЭК 8824-1-2001 «Информационная технология. Абстрактная синтаксическая нотация версии один (ASN.1). Часть 1. Спецификация основной нотации» (принят и введен в действие постановлением Госстандарта России от 6 сентября 2001 г. № 375-ст, опубликован в ноябре 2001 г. ИПК «Издательство стандартов», ИУС 11-2001).

<sup>3</sup> ГОСТ Р 34.11-2012 «Информационная технология. Криптографическая защита информации. Функция хэширования» (утвержден и введен в действие приказом Федерального агентства по техническому регулированию и метрологии от 7 августа 2012 г. № 216-ст, опубликован в апреле 2013 г. ФГУП «СТАНДАРТИНФОРМ», ИУС 3-2013, поправка ИУС 6-2018).

для алгоритма с длиной выхода 256 бит:

```
id-tc26-gost3411-12-256 OBJECT IDENTIFIER ::= { iso(1) member-body(2) ru(643)
rosstandart(7) tc26(1) algorithms(1) digest(2) gost3411-12-256(2) }
```

для алгоритма с длиной выхода 512 бит:

```
id-tc26-gost3411-12-512 OBJECT IDENTIFIER ::= { iso(1) member-body(2)
ru(643) rosstandart(7) tc26(1) algorithms(1) digest(2) gost3411-12-512(3) }
```

5.3. Поле `encapContentInfo` (тип `EncapsulatedContentInfo`) содержит подписываемые данные (`eContent`) вместе с их типом (`eContentType`) и определяется следующим образом:

```
EncapsulatedContentInfo ::= SEQUENCE {
    eContentType          ContentType,
    eContent              [0]      EXPLICIT OCTET STRING OPTIONAL }

ContentType ::= OBJECT IDENTIFIER
```

В случае если поле `eContent` присутствует, то в нем содержится подписываемый электронный документ. Если поле `eContent` отсутствует, электронный документ хранится в отдельном файле.

5.4. В поле `certificates` (тип `CertificateSet`) может включаться дополнительная информация о сертификатах. В данное поле может быть включена информация о сертификатах подписывающих сторон.

```
CertificateSet ::= SET OF CertificateChoices
```

```
CertificateChoices ::= CHOICE {
    certificate          Certificate,
    extendedCertificate [0]      IMPLICIT ExtendedCertificate,
    v1AttrCert          [1]      IMPLICIT AttributeCertificateV1,
    v2AttrCert          [2]      IMPLICIT AttributeCertificateV2,
    other               [3]      IMPLICIT OtherCertificateFormat }
```

5.4.1. Основной синтаксис типа `Certificate` представлен следующим образом:

```
Certificate ::= SEQUENCE {
    tbsCertificate      TBSCertificate,
    signatureAlgorithm  AlgorithmIdentifier,
    signatureValue      BIT STRING }
```

Поле `signatureAlgorithm` (тип `SignatureAlgorithmIdentifier`) определяет идентификатор использованного алгоритма ЭП и связанные с ним параметры:

```
SignatureAlgorithmIdentifier ::= AlgorithmIdentifier
```

В настоящем Формате в качестве идентификатора алгоритма ЭП указывается объектный идентификатор алгоритма, определенного ГОСТ Р 34.10-2012 «Информационная технология. Криптографическая защита информации. Процессы

формирования и проверки электронной цифровой подписи»<sup>1</sup> (далее – ГОСТ Р 34.10-2012). Указанный объектный идентификатор в соответствии со спецификацией абстрактной синтаксической нотации версии один представляется следующим образом:

для алгоритма с длиной выхода 256 бит:

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

для алгоритма с длиной выхода 512 бит:

```
id-tc26-gost3410-12-512 OBJECT IDENTIFIER ::= { iso(1) member-body(2)
ru(643) rosstandart(7) tc26(1) algorithms(1) sign(1) gost3410-2012-512(2) }
```

5.4.2. Поле `extendedCertificate` (типа `ExtendedCertificate`) определяет расширенный сертификат PKCS #6. Расширенные сертификаты PKCS #6 поддерживаются для совместимости с предыдущими версиями и не применяются.

5.4.3. Поле `v1AttrCert` (типа `AttributeCertificateV1`) определяет сертификат атрибута X.509 версии 1 в соответствии с пунктом 9 Требований к форме квалифицированного сертификата ключа проверки электронной подписи, утвержденных приказом ФСБ России от 27.12.2011 № 795 (зарегистрирован Минюстом России 27 января 2012 г., регистрационный № 23041), используется для совместимости с предыдущими версиями и применению не подлежит.

5.4.4. Поле `v2AttrCert` (типа `AttributeCertificateV2`) определяет сертификат атрибута X.509 версии 2 в соответствии с пунктом 9 Требований к форме квалифицированного сертификата ключа проверки электронной подписи, утвержденных приказом ФСБ России от 27.12.2011 № 795:

```
AttributeCertificateV2 ::= AttributeCertificate
```

5.4.5. Поле `other` (типа `OtherCertificateFormat`) предназначено для обеспечения возможности поддержки иных форматов сертификатов без внесения изменений в синтаксис криптографических сообщений:

```
OtherCertificateFormat ::= SEQUENCE {
  otherCertFormat OBJECT IDENTIFIER,
  otherCert ANY DEFINED BY otherCertFormat }
```

5.5 В поле `crls` (тип `RevocationInfoChoices`) может быть включена дополнительная информация об отзыве сертификатов:

```
RevocationInfoChoices ::= SET OF RevocationInfoChoice
```

```
RevocationInfoChoice ::= CHOICE {
```

<sup>1</sup> ГОСТ Р 34.10-2012 «Информационная технология. Криптографическая защита информации. Процессы формирования и проверки электронной цифровой подписи» (утвержден и введен в действие приказом Федерального агентства по техническому регулированию и метрологии от 7 августа 2012 г. № 215-ст, опубликован в апреле 2013 г. ФГУП «СТАНДАРТИНФОРМ», ИУС 3-2013, переиздание сентябрь 2018 г.).

```

crl                CertificateList,
other               [1]  IMPLICIT OtherRevocationInfoFormat }

```

```

OtherRevocationInfoFormat ::= SEQUENCE {
  otherRevInfoFormat  OBJECT IDENTIFIER,
  otherRevInfo        ANY DEFINED BY otherRevInfoFormat }

```

5.5.1. Поле `crl` (типа `CertificateList`) определяет список аннулированных сертификатов (CRL). Для вычисления подписи данные, которые должны быть подписаны, представляются в ASN.1-коде по DER-правилам:

```

CertificateList ::= SEQUENCE {
  tbsCertList      TBSCertList,
  signatureAlgorithm AlgorithmIdentifier,
  signatureValue   BIT STRING }

```

```

TBSCertList ::= SEQUENCE {
  version          Version OPTIONAL,
                  -- если представлено, то должно быть 2-й версии
  signature        AlgorithmIdentifier,
  issuer           Name,
  thisUpdate       Time,
  nextUpdate       Time OPTIONAL,
  revokedCertificates SEQUENCE OF SEQUENCE {
    userCertificate      CertificateSerialNumber,
    revocationDate      Time,
    crlEntryExtensions  Extensions OPTIONAL
                  -- если представлено, то должно быть 2-й версии
  } OPTIONAL,
  crlExtensions      [0] EXPLICIT Extensions OPTIONAL
                  -- если представлено, то должно быть 2-й версии
}

```

5.5.2. Поле `other` (типа `OtherRevocationInfoFormat`) определяет форматы информации об отзыве без дополнительного изменения синтаксиса криптографических сообщений.

5.6. В поле `signerInfos` (тип `SignerInfos`) содержится информация о каждой подписывающей стороне электронного документа.

```

SignerInfos ::= SET OF SignerInfo

```

```

SignerInfo ::= SEQUENCE {
  version          CMSVersion,
  sid              SignerIdentifier,
  digestAlgorithm  DigestAlgorithmIdentifier,
  signedAttrs      [0] IMPLICIT SignedAttributes OPTIONAL,
  signatureAlgorithm SignatureAlgorithmIdentifier,
  signature        SignatureValue,
  unsignedAttrs    [1] IMPLICIT UnsignedAttributes OPTIONAL }

```

5.6.1. Поле `sid` (тип `SignerIdentifier`) определяет сертификат подписывающей стороны, необходимый для проверки подлинности ЭП.

```

SignerIdentifier ::= CHOICE {
  issuerAndSerialNumber IssuerAndSerialNumber,
  subjectKeyIdentifier  [0] SubjectKeyIdentifier }

```

В структуре ЭП должен использоваться вариант определения сертификата путем задания значения `issuerAndSerialNumber`.

5.6.2. Поле `digestAlgorithm` (тип `DigestAlgorithmIdentifier`) определяет идентификаторы использованного в данной ЭП алгоритма хэширования и связанные с ним параметры.

В качестве идентификатора алгоритма хэширования указывается объектный идентификатор алгоритма, определенного ГОСТ Р 34.11-2012. Указанный объектный идентификатор в соответствии со спецификацией абстрактной синтаксической нотации версии один определяется следующим образом:

для алгоритма с длиной выхода 256 бит:

```
id-tc26-gost3411-12-256 OBJECT IDENTIFIER ::= { iso(1) member-body(2) ru(643)
rosstandart(7) tc26(1) algorithms(1) digest(2) gost3411-12-256(2) }
```

для алгоритма с длиной выхода 512 бит:

```
id-tc26-gost3411-12-512 OBJECT IDENTIFIER ::= { iso(1) member-body(2)
ru(643) rosstandart(7) tc26(1) algorithms(1) digest(2) gost3411-12-512(3) }
```

5.6.3. Поле `signedAttrs` (тип `SignedAttributes`) должно определять дополнительную подписываемую информацию (далее – подписываемые атрибуты). Обязательные к включению подписываемые атрибуты определяются следующим образом:

```
SignedAttributes ::= SET SIZE (1..MAX) OF Attribute
```

```
Attribute ::= SEQUENCE {
    attrType          OBJECT IDENTIFIER,
    attrValues        SET OF AttributeValue }
```

```
AttributeValue ::= ANY
```

5.6.4. Поле `signature` (тип `SignatureValue`) содержит строку бит, полученную в результате процесса формирования ЭП:

```
SignatureValue ::= OCTET STRING
```

5.6.5. Поле `unsignedAttrs` (тип `UnsignedAttributes`) определяет дополнительную неподписываемую информацию (далее – неподписываемые атрибуты):

```
UnsignedAttributes ::= SET SIZE (1..MAX) OF Attribute
```

6. Обязательными подписываемыми атрибутами являются:

6.1. Тип содержимого (`Content-type`). Должен быть добавлен в атрибут `id-contentType` с объектным идентификатором вида «1.2.840.113549.1.3»;

6.2. Строка бит, являющаяся выходным результатом функции, отображающей строки бит в строки бит фиксированной длины и удовлетворяющей следующим условиям:

по данному значению функции невозможно на период действия сертификата вычислить исходные данные, отображаемые в этом значении;

для заданных исходных данных невозможно на период действия сертификата вычислить другие исходные данные, отображаемые в этом значении;

невозможно на период действия сертификата вычислить какую-либо пару исходных данных, отображаемых в одно и то же значение хэш-функции (далее – хэш-код) электронного сообщения (Message-digest).

Результат функции, отображающей строки бит в строки бит фиксированной длины и удовлетворяющей указанным условиям должен быть добавлен в атрибут `id-messageDigest` с объектным идентификатором вида «1.2.840.113549.1.4»;

6.3. Хэш-код от набора полей сертификата, позволяющих однозначно его идентифицировать, должен быть добавлен в атрибут `id-aa-signingCertificateV2` с объектным идентификатором вида «1.2.840.113549.1.9».

7. В случае если участник электронного взаимодействия не является владельцем действующего сертификата ключа проверки электронной подписи, Формат должен иметь вид следующей структуры:

```
CertificationRequest ::= SEQUENCE {
    certificationRequestInfo CertificationRequestInfo,
    signatureAlgorithm      AlgorithmIdentifier {{Signature Algorithms}},
    signature                BITSTRING}, где
```

`certificationRequestInfo` – подписываемая информация;

`signatureAlgorithm` – информация об алгоритме ЭП, который использовался при формировании ЭП структуры `certificationRequestInfo`;

`signature` – ЭП структуры `certificationRequestInfo`, сформированная с использованием алгоритма, указанного в `signatureAlgorithm`.

7.1. Структура `CertificationRequestInfo` в соответствии со спецификацией абстрактной синтаксической нотации версии один представляется следующим образом:

```
CertificationRequestInfo ::= SEQUENCE {
    version          INTEGER { v1(0) } (v1,...),
    subject          Name,
    subjectPKInfo   SubjectPublicKeyInfo{{ PKInfoAlgorithms }},
    attributes [0] Attributes{{ CRIAttributes }}}, где
```

`version` – номер версии стандарта (в данном формате данное поле должно принимать значение 0)

`subject` – имя субъекта, владеющего ключом ЭП, имеющее тип `Name`, которое заполняется в соответствии с Требованиями к форме квалифицированного сертификата ключа проверки электронной подписи, утвержденными приказом ФСБ России от 27.12.2011 г. № 795;

`subjectPKInfo` – набор элементов, содержащих информацию о ключе проверки ЭП и имеющих структуру `SubjectPublicKeyInfo`;

`attributes` – набор атрибутов.

Структура `SubjectPublicKeyInfo` в соответствии со спецификацией абстрактной синтаксической нотации версии один представляется следующим образом:

```
SubjectPublicKeyInfo (ALGORITHM: IOSet) ::= SEQUENCE {
    algorithm      AlgorithmIdentifier {{IOSet}}
```



`subjectPublicKey` BIT STRING }, где

`algorithm` – алгоритм и набор параметров алгоритма ЭП;

`subjectPublicKey` – ключ проверки ЭП.

Поле `algorithm` структуры `SubjectPublicKeyInfo` задается структурой `AlgorithmIdentifier`, описанной в ГОСТ Р ИСО/МЭК 9594-8 и содержащей следующие поля:

`algorithm` – идентификатор алгоритма ЭП;

`parameters` – параметры открытого ключа алгоритма ЭП.

Поле `algorithm` структуры `AlgorithmIdentifier` должно содержать следующий идентификатор алгоритма:

для алгоритма ГОСТ Р 34.10-2012 с длиной ключа 256 бит идентификатор в соответствии со спецификацией абстрактной синтаксической нотации версии один представляется следующим образом:

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

для алгоритма ГОСТ Р 34.10-2012 с длиной ключа 512 бит идентификатор в соответствии со спецификацией абстрактной синтаксической нотации версии один представляется следующим образом:

```
id-tc26-gost3410-12-512 OBJECT IDENTIFIER ::=
{ iso(1) member-body(2) ru(643) rosstandart (7) tc26(1) algorithms(1)
sign(1) gost3410-2012-512(2) }
```

Поле `parameters` структуры `AlgorithmIdentifier` имеет структуру `GostR3410-2012-PublicKeyParameters`, которая в соответствии со спецификацией абстрактной синтаксической нотации версии один представляется следующим образом:

```
GostR3410-2012-PublicKeyParameters ::= SEQUENCE {
publicKeyParamSet OBJECT IDENTIFIER,
digestParamSet OBJECT IDENTIFIER OPTIONAL }, где
```

`publicKeyParamSet` – идентификатор параметров ключа проверки ЭП, соответствующего алгоритму ЭП ГОСТ Р 34.10-2012;

`digestParamSet` – идентификатор хэш-функции ГОСТ Р 34.11-2012.

данное поле не следует использовать, если используется алгоритм ЭП ГОСТ Р 34.10-2012 с длиной ключа 512 бит;

данное поле должно присутствовать, если используется алгоритм ЭП ГОСТ Р 34.10-2012 с длиной ключа 256 бит при следующих значениях поля `publicKeyParamSet`:

```
id-GostR3410-2001-CryptoPro-A-ParamSet
id-GostR3410-2001-CryptoPro-B-ParamSet
id-GostR3410-2001-CryptoPro-C-ParamSet
id-GostR3410-2001-CryptoPro-XchA-ParamSet
id-GostR3410-2001-CryptoPro-XchB-ParamSet
```

при этом он должен быть равен идентификатору алгоритма хэширования ГОСТ Р 34.11-2012 с длиной выхода 256 бит:

```
id-tc26-gost3411-12-256
```

данное поле не следует использовать, если используется алгоритм ЭП ГОСТ Р 34.10-2012 с длиной ключа 256 бит при следующем значении поля `publicKeyParamSet`:

```
id-tc26-gost-3410-2012-256-paramSetA
```

данное поле должно отсутствовать, если используется алгоритм ЭП ГОСТ Р 34.10-2012 с длиной ключа 256 бит при следующих значениях поля `publicKeyParamSet`:

```
id-tc26-gost-3410-2012-256-paramSetB
id-tc26-gost-3410-2012-256-paramSetC
id-tc26-gost-3410-2012-256-paramSetD
```

Поле `subjectPublicKey` структуры `SubjectPublicKeyInfo` содержит ключ проверки ЭП, который задается парой координат (x,y), определенной ГОСТ Р 34.10-2012, представляется следующим образом:

ключ проверки ЭП, соответствующий алгоритму ГОСТ Р 34.10-2012 с длиной ключа 256 бит, имеет представление `GostR3410-2012-256-PublicKey`, которое задается байтовой строкой длины 64 байта, где первые 32 байта содержат представление координаты x в формате `little-endian` (младший бит записывается первым), а последние 32 байта содержат представление координаты y в формате `little-endian`;

ключ проверки ЭП, соответствующий алгоритму ГОСТ Р 34.10-2012 с длиной ключа 512 бит, имеет представление `GostR3410-2012-512-PublicKey`, которое задается байтовой строкой длины 128 байт, где первые 64 байта содержат представление координаты x в формате `little-endian`, а последние 64 байта содержат представление координаты y в формате `little-endian`.

Ключи проверки ЭП `GostR3410-2012-256-PublicKey` и `GostR3410-2012-512-PublicKey` должны быть представлены в ASN.1-коде по DER-правилам в виде строки байт (OCTET STRING) в соответствии с ГОСТ Р ИСО/МЭК 8825-1:

```
GostR3410-2012-256-PublicKey ::= OCTET STRING (64)
GostR3410-2012-512-PublicKey ::= OCTET STRING (128)
```

7.2. Поле `algorithm` структуры `signatureAlgorithm` содержит идентификатор алгоритма ЭП и связанных с ним параметров, который используется при формировании ЭП структуры `CertificationRequestInfo` с использованием ключа ЭП:

для алгоритма ГОСТ Р 34.10-2012 с длиной ключа 256 бит идентификатор в соответствии со спецификацией абстрактной синтаксической нотации версии один представляется следующим образом:

```
id-tc26-signwithdigest-gost3410-12-256 OBJECT IDENTIFIER ::=
{ iso(1) member-body(2) ru(643) rosstandart(7) tc26(1) algorithms(1)
  signwithdigest(3) gost3410-2012-256(2) }
```

для алгоритма ГОСТ Р 34.10-2012 с длиной ключа 512 бит идентификатор в соответствии со спецификацией абстрактной синтаксической нотации версии один представляется следующим образом:

```
id-tc26-signwithdigest-gost3410-12-512 OBJECT IDENTIFIER ::=
{iso(1) member-body(2) ru(643) rosstandart(7) tc26(1) algorithms(1)
signwithdigest(3) gost3410-2012-512(3)}
```

Поле `parameters` структуры `signatureAlgorithm` должно отсутствовать.

7.3. Поле `signature` структуры `CertificationRequest` содержит ЭП подписываемой информации.

Результатом работы алгоритма ЭП ГОСТ Р 34.10-2012 с длиной ключа 256 бит является два числа `r` и `s`, определенные ГОСТ Р 34.10-2012 и имеющие длину 256 бит каждое.

При использовании алгоритма ГОСТ Р 34.10-2012 с длиной ключа 256 бит поле `signature` задается битовой строкой (BITSTRING) длины 512 бит; при этом первые 256 бит содержат число `s` в представлении `big-endian` (старший бит записывается первым), а вторые 256 бит содержат число `r` в представлении `big-endian`.

Результатом работы алгоритма ЭП ГОСТ Р 34.10-2012 с длиной ключа 512 бит является два числа `r` и `s`, определенные ГОСТ Р 34.10-2012 и имеющие длину 512 бит каждое.

При использовании алгоритма ГОСТ Р 34.10-2012 с длиной ключа 512 бит поле `signature` задается битовой строкой (BITSTRING) длины 1024 бит; при этом первые 512 бит содержат число `s` в представлении `big-endian`, а вторые 512 бит содержат число `r` в представлении `big-endian`.

---