МИНИСТЕРСТВО ЦИФРОВОГО РАЗВИТИЯ, СВЯЗИ И МАССОВЫХ КОММУНИКАЦИЙРОССИЙСКОЙ ФЕДЕРАЦИИ

(МИНКОМСВЯЗЬ РОССИИ)

ПРИКАЗ

_____ № _____

Москва

Об утверждении порядка создания метки доверенного времени

В соответствии с пунктом 19 статьи 2 Федерального закона от 6 апреля 2011 г. № 63-ФЗ «Об электронной подписи» (Собрание законодательства, 2011, № 15, ст. 2036; 2019, № 52, ст. 7794)

ПРИКАЗЫВАЮ:

1. Утвердить прилагаемый порядок создания метки времени.
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 _____ № _____

Порядок создания, проверки и получения информации о дате и времени подписания электронного документа электронной подписью

1. Настоящий Порядок определяет правила создания, проверки и получения достоверной информации в электронной форме о дате и времени подписания электронного документа в момент его подписания электронной подписью (далее – метка доверенного времени).

2. Метка доверенного времени создается доверенной третьей стороной, удостоверяющим центром или оператором информационной системы (далее – служба меток доверенного времени) с использованием программных и (или) аппаратных средств, прошедших процедуру подтверждения соответствия требованиям, установленным в соответствии с частью 5 статьи 8 Федерального закона от 6 апреля 2011 г. № 63-ФЗ «Об электронной подписи» (далее – Федеральный закон «Об электронной подписи).

3. Создание метки доверенного времени осуществляется службой меток доверенного времени по запросам путем подписания электронной подписью службы меток доверенного времени текущего достоверного значения времени в соответствии с Приложением 2 настоящего Приказа.

4. Метка доверенного времени может быть присоединена к подписываемому документу с целью обеспечения достоверной информации о моменте подписания электронного документа или связана с подписываемым документом иным способом.

5. Формат запроса метки доверенного времени и формат ответа службы меток доверенного времени на такой запрос, предусматривающий обеспечение целостности и достоверности метки доверенного времени определяется службой меток доверенного времени в соответствии с международными и национальными стандартами, а также иными документами по стандартизации в соответствии с законодательством Российской Федерации в сфере технического регулирования и стандартизации.

6. Службы меток доверенного времени, являющиеся компонентами доверенной третьей стороны, удостоверяющего центра, при создании метки доверенного времени должны получать информацию о точном значении московского времени и календарной дате от технических средств передачи эталонных сигналов времени и частоты, а также информации о точном значении московского времени и календарной дате (далее – технические средства передачи эталонных сигналов времени), функционирующих в соответствии с Положением о Государственной службе времени, частоты и определения параметров вращения Земли, утвержденным Постановлением Правительства Российской Федерации

от 23 марта 2001 г. № 225 (Собрание законодательства Российской Федерации, 2001, № 14, ст. 1361; 2018, № 49, ст. 7600).

7. Службы меток доверенного времени, являющиеся компонентами операторов информационных систем, могут получать информацию о дате и времени от технических средств передачи эталонных сигналов времени или от службы меток доверенного времени, указанных в пункте 5 настоящего Порядка посредством информационно-телекоммуникационных сетей по протоколу сетевого времени при условии обеспечения целостности передаваемой информации с использованием средств криптографической защиты информации, имеющих подтверждение соответствия требованиям ФСБ России.

8. Проверка метки доверенного времени осуществляется доверенной третьей стороной, удостоверяющим центром или оператором информационных систем при проверке действительности электронной подписи с использованием средств, прошедших процедуру подтверждения соответствия требованиям, установленным в соответствии с частью 5 статьи 8 Федерального закона «Об электронной подписи».

Приложение к приказу
Министерства цифрового развития,
связи и массовых коммуникаций
Российской Федерации
от _____ № _____

Требования к структуре метки доверенного времени

I. ОБЩИЕ СВЕДЕНИЯ

1. Настоящие Требования определяют структуру информации о дате и времени подписания электронного документа.

II. ОПИСАНИЕ СТРУКТУРЫ ЗАЩИЩЕННЫХ ДАННЫХ

2. Структура метки доверенного времени представляет собой защищенные данные (ContentInfo), которые могут содержать один из следующих типов содержимого:

- простые данные (it-data);
- подписанные данные (signed-data);
- конверт данных (enveloped-data);
- хэшированные данные (digested-data);
- зашифрованные данные (encrypted-data);
- аутентифицированные данные (authenticated-data).

3. ContentInfo инкапсулирует один тип содержимого, данный тип также может быть подвергнут дальнейшей инкапсуляции.

4. Следующий идентификатор определяет тип содержимого:

id-ct-ContentInfo OBJECT IDENTIFIER ::=
{ iso(1) member-body(2) us(840) rsadsi(113549) pkcs(1) pkcs9(9) smime(16) ct(1) 6
}

Формат CMS связывает идентификатор типа содержимого с самим содержимым.

Тип ContentInfo в формате АСН.1 представлен следующим образом:

```
ContentInfo ::= SEQUENCE
{
        contentType             ContentType,
        content[0]              EXPLICIT ANY DEFINED BY contentType
}
```

ContentType ::= OBJECT IDENTIFIER

Поля структуры ContentInfo имеют следующий смысл:

- contentType – тип соответствующего содержимого;

- content – соответствующее содержимое. Тип содержимого может быть однозначно определен по идентификатору в поле contentType.

5. Содержимое типа «простые данные» определено следующим идентификатором:

id-data OBJECT IDENTIFIER ::=
{ iso(1) member-body(2) us(840) rsadsi(113549) pkcs(1)
pkcs7(7) 1
}

Тип содержимого «простые данные» предназначен для обозначения произвольной строки байтов, например, текстовые файлы ASCII. Такие строки не должны иметь никакой значимой для CMS формата внутренней структуры.

6. Содержимое типа «подписанные данные» представляет собой данные любого типа и любое количество подписей. Любое количество отправителей, осуществляющих подпись, могут подписывать произвольное содержимое независимо друг от друга.

7. Содержимое типа «конверт данных» представляет собой зашифрованное содержимое и зашифрованные ключи шифрования содержимого для одного получателя или более.

Значение RecipientInfo вместе с зашифрованным содержимым для всех получателей записывают в структуру EnvelopedData.

Идентификатор, который определяет тип содержимого EnvelopedData:

id-envelopedData OBJECT IDENTIFIER ::=
{ iso(1) member-body(2) us(840) rsadsi(113549) pkcs(1)
pkcs7(7) 3
}

Тип EnvelopedData в формате АСН.1 представляется следующим образом:

EnvelopedData ::= SEQUENCE
{
version CMSVersion,
originatorInfo[0] IMPLICIT OriginatorInfo
OPTIONAL,
recipientInfos RecipientInfos,
encryptedContentInfo EncryptedContentInfo,
unprotectedAttrs[1] IMPLICIT UnprotectedAttributes
OPTIONAL
}
RecipientInfos ::= SET SIZE (1..MAX) OF RecipientInfo
EncryptedContentInfo ::= SEQUENCE
{
contentType ContentType,
contentEncryptionAlgorithm
ContentEncryptionAlgorithmIdentifier, encryptedContent[0]
IMPLICIT EncryptedContent OPTIONAL
}

ContentType ::= OBJECT IDENTIFIER
ContentEncryptionAlgorithmIdentifier ::= SEQUENCE
{ encryptionAlgorithmOID OBJECT IDENTIFIER, parameters Gost3412-15-Encryption-Parameters
}
Gost3412-15-Encryption-Parameters ::= SEQUENCE
{ ukm OCTET STRING
}
EncryptedContent ::= OCTET STRING
UnprotectedAttributes ::= SET SIZE (1..MAX) OF Attribute

Для каждого получателя зашифрованный ключ шифрования содержимого и другая информация, специфичная для получателя, записывают в структуру RecipientInfo:

RecipientInfo ::= CHOICE
{
ktri KeyTransRecipientInfo, kari[1] KeyAgreeRecipientInfo, kekri[2] KEKRecipientInfo, pwri[3] PasswordRecipientinfo, ori[4] OtherRecipientInfo
}

8. Содержимое типа «хэшированные данные» состоит из содержимого любого типа и результата функции хэширования содержимого.
Идентификатор, который определяет тип «хэшированные данные»:

id-digestedData OBJECT IDENTIFIER ::=
{ iso(1) member-body(2) us(840) rsadsi(113549) pkcs(1) pkcs7(7) 5
}
Содержимое типа «хэшированные данные» представлено в виде структуры DigestedData:
DigestedData ::= SEQUENCE
{
version CMSVersion,
digestAlgorithm DigestAlgorithmIdentifier,
encapContentInfo EncapsulatedContentInfo,
digest Digest
}
DigestAlgorithmIdentifier ::= AlgorithmIdentifier
Digest ::= OCTET STRING

9. Содержимое типа «зашифрованные данные» состоит из зашифрованного содержимого любого типа. В отличие от содержимого типа

«конверт данных» данный тип не содержит ни получателей, ни зашифрованных ключей. Распределение ключей происходит с помощью внешних средств.

Идентификатор, который определяет тип «зашифрованные данные»:

id-encryptedData OBJECT IDENTIFIER ::=

{ iso(1) member-body(2) us(840) rsadsi(113549) pkcs(1) pkcs7(7) 6 }

Содержимое типа «зашифрованные данные» представлено в виде структуры EncryptedData:

```
EncryptedData ::= SEQUENCE
{
        version                 CMSVersion,
                                EncryptedContentInfo,
encryptedContentInfo
 unprotectedAttrs[1]  OPTIONAL          IMPLICIT
}                               UnprotectedAttributes

EncryptedContentInfo ::= SEQUENCE
{
        contentType                     ContentType,
          contentEncryptionAlgorithm
ContentEncryptionAlgorithmIdentifier,   encryptedContent[0]
IMPLICIT EncryptedContent OPTIONAL
}
ContentType ::= OBJECT IDENTIFIER
ContentEncryptionAlgorithmIdentifier ::= SEQUENCE
{  encryptionAlgorithmOID  OBJECT  IDENTIFIER,  parameters
Gost3412-15-Encryption-
Parameters
}
Gost3412-15-Encryption-Parameters ::= SEQUENCE
{       ukm OCTET STRING
}
EncryptedContent ::= OCTET STRING
UnprotectedAttributes ::= SET SIZE (1..MAX) OF Attribute .
```

10. Содержимое типа «аутентифицированные данные» состоит из содержимого любого типа, имитовставки, зашифрованных ключей имитозащиты для одного получателя или более.

Сочетание имитовставки и зашифрованного ключа имитозащиты необходимо для получателя, для того чтобы проверить целостность содержимого. Любое содержимое может быть защищено для любого количества получателей.

Идентификатор, который определяет данные типа «аутентифицированные данные»: id-ct-authData OBJECT IDENTIFIER ::=

{ iso(1) member-body(2) us(840) rsadsi(113549) pkcs(1) pkcs-9(9) smime(16) ct(1) 2
}

Содержимое типа «аутентифицированные данные» представлено в виде структуры AuthenticatedData:

AuthenticatedData ::= SEQUENCE
{
version CMSVersion,
originatorInfo[0] IMPLICIT OriginatorInfo OPTIONAL, recipientInfos RecipientInfos, macAlgorithm MessageAuthenticationCodeAlgorithm, digestAlgorithm[1] DigestAlgorithmIdentifier OPTIONAL, encapContentInfo EncapsulatedContentInfo, authAttrs[2] IMPLICIT AuthAttributes OPTIONAL, mac MessageAuthenticationCode,
unauthAttrs[3] IMPLICIT UnauthAttributes OPTIONAL
}

RecipientInfos ::= SET SIZE (1..MAX) OF RecipientInfo
MessageAuthenticationCodeAlgorithm ::= AlgorithmIdentifier
DigestAlgorithmIdentifier ::= AlgorithmIdentifier
AuthAttributes ::= SET SIZE (1..MAX) OF Attribute
UnauthAttributes ::= SET SIZE (1..MAX) OF Attribute
MessageAuthenticationCode ::= OCTET STRING