
ФЕДЕРАЛЬНОЕ АГЕНТСТВО
ПО ТЕХНИЧЕСКОМУ РЕГУЛИРОВАНИЮ И МЕТРОЛОГИИ



РЕКОМЕНДАЦИИ
ПО СТАНДАРТИЗАЦИИ

**Р 50.1.112—
2016**

Информационная технология
КРИПТОГРАФИЧЕСКАЯ ЗАЩИТА ИНФОРМАЦИИ
Транспортный ключевой контейнер

Издание официальное



Москва
Стандартинформ
2016

Предисловие

1 РАЗРАБОТАНЫ подкомитетом 2 Технического комитета по стандартизации ТК 26 «Криптографическая защита информации»

2 ВНЕСЕНЫ Техническим комитетом по стандартизации ТК 26 «Криптографическая защита информации»

3 УТВЕРЖДЕНЫ И ВВЕДЕНЫ В ДЕЙСТВИЕ Приказом Федерального агентства по техническому регулированию и метрологии от 23 ноября 2016 г. № 1753-ст

4 ВВЕДЕНЫ ВПЕРВЫЕ

Правила применения настоящего стандарта установлены в статье 26 Федерального закона от 29 июня 2015 г. № 162-ФЗ «О стандартизации в Российской Федерации». Информация об изменениях к настоящим рекомендациям публикуется в ежегодном (по состоянию на 1 января текущего года) информационном указателе «Национальные стандарты», а официальный текст изменений и поправок — в ежемесячном информационном указателе «Национальные стандарты». В случае пересмотра (замены) или отмены настоящих рекомендаций соответствующее уведомление будет опубликовано в ближайшем выпуске ежемесячного информационного указателя «Национальные стандарты». Соответствующая информация, уведомление и тексты размещаются также в информационной системе общего пользования — на официальном сайте Федерального агентства по техническому регулированию и метрологии в сети Интернет (www.gost.ru)

© Стандартиформ, 2016

Настоящие рекомендации не могут быть полностью или частично воспроизведены, тиражированы и распространены в качестве официального издания без разрешения Федерального агентства по техническому регулированию и метрологии

Содержание

1 Область применения	1
2 Нормативные ссылки	1
3 Обозначения	1
3.1 Обозначения	1
4 Представление ключей ГОСТ Р 34.10 и ГОСТ 28147—89	2
4.1 Портфель ключевой информации KeyBag	2
4.2 Данные, защищаемые в транспортном ключевом контейнере	3
4.3 Обеспечение целостности и конфиденциальности ключей	3
5 Парольная защита	3
6 Выработка ключа по протоколу Диффи-Хелмана	4
7 Формат PFX контейнера	4
8 Модули ASN.1	5
Приложение А (справочное) Контрольные примеры	6
Библиография	20

Введение

Настоящие рекомендации содержат расширения документов PKCS#8 «PrivateKey Information Syntax Standard» версии 1.2 [1] и PKCS#12 «Personal Information Exchange Syntax» версии 1.0 [2], описывающие формирование транспортных ключевых контейнеров для ключей, созданных в соответствии с ГОСТ Р 34.10.

Необходимость разработки настоящих рекомендаций вызвана необходимостью разработки решения, использующего национальные криптографические стандарты, для обеспечения безопасной передачи ключевой информации.

П р и м е ч а н и е — Основная часть настоящих рекомендаций дополнена приложением А.

Информационная технология
КРИПТОГРАФИЧЕСКАЯ ЗАЩИТА ИНФОРМАЦИИ
Транспортный ключевой контейнер

Information technology. Cryptographic data security. Transport key container

Дата введения — 2017—06—01

1 Область применения

Настоящие рекомендации предназначены для применения в информационных системах, использующих механизмы электронной подписи по ГОСТ Р 34.10 в общедоступных и корпоративных сетях для защиты информации, не содержащей сведений, составляющих государственную тайну.

2 Нормативные ссылки

В настоящих рекомендациях использованы нормативные ссылки на следующие стандарты:

ГОСТ 28147—89 Системы обработки информации. Защита криптографическая. Алгоритмы криптографического преобразования

ГОСТ Р 34.10—2012 Информационная технология. Криптографическая защита информации. Процессы формирования и проверки электронной цифровой подписи

ГОСТ Р 34.11—2012 Информационная технология. Криптографическая защита информации. Функция хэширования

Р 50.1.113 — 2016 Информационная технология. Криптографическая защита информации. Криптографические алгоритмы, сопутствующие применению алгоритмов электронной цифровой подписи и функции хэширования

Р 50.1.111— 2016 Информационная технология. Криптографическая защита информации. Парольная защита ключевой информации

П р и м е ч а н и е — При пользовании настоящими рекомендациями целесообразно проверить действие ссылочных стандартов в информационной системе общего пользования — на официальном сайте Федерального агентства по техническому регулированию и метрологии в сети Интернет или по ежегодному информационному указателю «Национальные стандарты», который опубликован по состоянию на 1 января текущего года, и по выпускам ежемесячного информационного указателя «Национальные стандарты» за текущий год. Если заменен ссылочный стандарт, на который дана недатированная ссылка, то рекомендуется использовать действующую версию этого стандарта с учетом всех внесенных в данную версию изменений. Если заменен ссылочный стандарт, на который дана датированная ссылка, то рекомендуется использовать версию этого стандарта с указанным выше годом утверждения (принятия). Если после утверждения настоящих рекомендаций в ссылочный стандарт (рекомендации), на который дана датированная ссылка, внесено изменение, затрагивающее положение, на которое дана ссылка, то это положение рекомендуется применять без учета данного изменения. Если ссылочный стандарт (рекомендации) отменен без замены, то положение, в котором дана ссылка на него, применяется в части, не затрагивающей эту ссылку.

3 Обозначения

3.1 Обозначения

P — пароль, представляющий собой символьную строку в кодировке Unicode UTF-8;

S — случайное значение синхропосылки;

\parallel — конкатенация двух байтовых строк; для двух байтовых строк $\alpha = (\alpha_1, \dots, \alpha_{n_1}) \in B_{n_1}$, $\beta = (\beta_1, \dots, \beta_{n_2}) \in B_{n_2}$ их конкатенацией $\alpha \parallel \beta$ называется байтовая строка $\gamma = (\alpha_1, \dots, \alpha_{n_1}, \beta_1, \dots, \beta_{n_2}) \in B_{n_1+n_2}$.

4 Представление ключей ГОСТ Р 34.10 и ГОСТ 28147—89

Для обеспечения защиты закрытых ключей от утечек по побочным каналам при считывании и проведении операций с ключами, целесообразно использование маскированных ключей. Для хранения маскированных ключей и наборов масок предлагаются следующие принципы.

Алгоритм наложения маски определен базовой операцией криптографического преобразования алгоритма. Для ключей по ГОСТ Р 34.10 это умножение в поле F_q , для ключей по ГОСТ 28147—89 — сложение по модулю 2^{32} .

Задана последовательность из k масок M_1, M_2, \dots, M_k . Через $M_i(\cdot)$ обозначают операцию наложения i -й маски, а через $M_i^{-1}(\cdot)$ — операцию снятия i -й маски, $1 \leq i \leq k$. Имеется ключ K . Маскированный ключ K_M получается в результате k -кратного применения операции наложения маски, а именно $K_M = M_k(\dots(M_2(M_1(K))))$. Демаскирование выполняется при помощи k -кратного применения операции снятия маски, но в обратном порядке, а именно $K = M_1^{-1}(\dots(M_{k-1}^{-1}(M_k^{-1}(K_M))))$. Маскированный ключ представляется как последовательность $I = K_M \| M_1 \| M_2 \| \dots \| M_k$, где « $\|$ » — операция конкатенации. Предположим, ключ K имеет n двоичных разрядов, тогда для представления в памяти последовательности I понадобится $(k+1)n$ двоичных разрядов или $k+1$ n -разрядных блоков.

Поскольку размерность типа INTEGER может изменяться при наличии нулевых значений в старших разрядах числа, использование такого типа для представления закрытого ключа создает неудобства при вычислении размерности.

Таким образом, пара ключей для алгоритма по ГОСТ Р 34.10 представлена в виде:

```
GostR3410-2012-KeyValueInfo ::= SEQUENCE{
  GostR3410-2012-KeyValueMask,
  GostR3410-2012-PublicKey }, где
```

```
GostR3410-2012-KeyValueMask ::= OCTET STRING {  $K_M \| M_1 \| M_2 \| \dots \| M_k$  }, и
GostR3410-2012-PublicKey ::= OCTET STRING { PubKeyX | PubKeyY }.
```

Для алгоритма ГОСТ 28147—89 ключ представляется в виде:

```
Gost28147-89-Key-KeyValueMask ::= OCTET STRING {  $K_M \| M_1 \| M_2 \| \dots \| M_k$  }
```

Возможно использование немаскированного закрытого ключа (т. е. $k = 0$, $K_M = K$).

Для обеспечения унификации между представлениями ключей в PFX и сертификатах формата X.509, как секретный, так и открытый ключи представляются в формате little-endian (старший байт справа).

Операцией наложения маски является умножение ключа на число, обратное маске:

$$K_M = K * M^{-1} \text{ mod } Q,$$

где значение Q взято из параметров ключа. Соответственно операцией снятия маски является умножение маскированного ключа на маску:

$$K = K_M * M \text{ mod } Q.$$

4.1 Портфель ключевой информации KeyBag

В соответствии с [1] и [2] портфель ключевой информации KeyBag представлен в следующем виде:

```
PrivateKeyInfo ::= SEQUENCE {
  version Version,
  privateKeyAlgorithm PrivateKeyAlgorithmIdentifier,
  privateKey PrivateKey,
  attributes [0] IMPLICIT Attributes OPTIONAL }
```

```
Version ::= INTEGER
```

Версия структуры на данный момент равна нулю.

```
PrivateKeyAlgorithmIdentifier ::= AlgorithmIdentifier
```

Для закрытых ключей по ГОСТ Р 34.10 используют идентификаторы соответствующих открытых ключей, представленные в рекомендациях [3].

```
PrivateKey ::= OCTET STRING
```

Содержимым данного типа является значение закрытого ключа, закодированное в соответствии с типом GostR3410-2012-PrivateKey.

```
GostR3410-2012-PrivateKey ::= CHOICE {
  GostR3410-2012-KeyValueMask,
  GostR3410-2012-KeyValueInfo }
Attributes ::= SET OF Attribute
```

Атрибуты могут содержать дополнительную необходимую информацию о ключе.
Ключ в зашифрованном виде представлен в виде структуры:

```
EncryptedPrivateKeyInfo ::= SEQUENCE {
  encryptionAlgorithm EncryptionAlgorithmIdentifier,
  encryptedData EncryptedData }
EncryptionAlgorithmIdentifier ::= AlgorithmIdentifier
```

При шифровании должен быть использован алгоритм PBES2 по P 50.1.111—2016. Алгоритм и параметры шифрования EncryptionAlgorithmIdentifier указаны в соответствии с P 50.1.111—2016.

```
EncryptedData ::= OCTET STRING
```

Содержимым данного типа является результат зашифрования кодированной структуры PrivateKeyInfo.

4.2 Данные, защищаемые в транспортном ключевом контейнере

В соответствии с 4.1 [2] каждый раздел транспортных ключевых контейнеров (далее — ТКК), содержащий конфиденциальные сведения, должен быть зашифрован в рамках ContentInfo типа AuthenticatedSafe. Портфель сертификата (CertBag — см. 4.2.3 [2]), соответствующий присутствующему в ТКК портфелю закрытого ключа (KeyBag), несет в себе информацию, облегчающую потенциальному злоумышленнику задачу первичного анализа перехваченного зашифрованного сообщения, в частности по данным, содержащимся в сертификате, можно получить информацию о владельце секретного ключа. В этой связи содержащийся в ТКК CertBag также может быть зашифрован.

Формирование парольного ключа для шифрования различных ContentInfo и портфелей закрытого ключа KeyBag должно быть осуществлено с использованием различных уникальных синхроросылок, что исключает повторное использование одного и того же секретного ключа для шифрования различных разделов ТКК.

4.3 Обеспечение целостности и конфиденциальности ключей

Для обеспечения целостности и конфиденциальности ключей используют транспортный ключ, выработанный одним из способов, перечисленных в следующих разделах.

5 Парольная защита

У отправителя и принимающего имеется предварительно согласованный пароль *P*. Отправитель с помощью алгоритма диверсификации вырабатывает парольный ключ по алгоритму PBKDF2 в соответствии с P 50.1.111—2016 и использует его для шифрования передаваемого закрытого ключа. Принимающий независимо вырабатывает парольный ключ и использует его для извлечения закрытого ключа из ТКК.

Целостность ТКК обеспечивается с использованием алгоритма HMAC_GOSTR3411_2012_512 в соответствии с P 50.1.113—2016.

Ввиду простоты реализации этот способ является наиболее приемлемым для большинства практических приложений.

Для шифрования различных разделов ТКК используется один и тот же пароль *P*, но различные случайные значения параметра *S* длиной от 8 до 32 байт.

Пароль должен быть представлен в формате UTF-8 без завершающего нуля и подан на вход алгоритма PBKDF2 в качестве параметра *P*.

При проверке целостности ТКК с помощью алгоритма HMAC_GOSTR3411_2012_512 ключ для данного алгоритма также вырабатывается по алгоритму PBKDF2 с тем же самым значением параметра *P* и случайным вектором *S* длиной от 8 до 32 байт. Ключом алгоритма HMAC_GOSTR3411_2012_512 должны быть последние 32 байта 96-байтовой последовательности, вырабатываемой с помощью PBKDF2.

Использован идентификатор алгоритма:

```
id-tc26-hmac-gost-3411-12-512 ::= { iso(1) member-body(2) ru(643) rosstandart (7) tk26(1) algorithms(1) mac(4) hash512(1) }
```

Функция HMAC_GOSTR3411_2012_512 вычисляется от содержимого поля content структуры authSafe (см. 7).

6 Выработка ключа по протоколу Диффи-Хелмана

В случае наличия у отправителя предварительно распределенной ключевой пары, сформированной по алгоритму ГОСТ Р 34.10 и соответствующего сертификата электронной подписи, для согласования ключей защиты ТКК должен быть использован алгоритм VKO_GOSTR3410_2012_256 или VKO_GOSTR3410_2012_512 в зависимости от параметров ключей (см. раздел 6 [4]).

Целостность контейнера в этом случае обеспечивается электронной подписью на ключе отправителя.

7 Формат PFX контейнера

В соответствии с PKCS#12 контейнер имеет вид:

```
PFX ::= SEQUENCE {
    version          INTEGER {v3(3)}(v3,...),
    authSafe         ContentInfo,
    macData          MacData OPTIONAL
}
MacData ::= SEQUENCE {
    mac              DigestInfo,
    macSalt          OCTET STRING,
    iterations       INTEGER DEFAULT 1
},
```

где authSafe в зависимости от метода контроля целостности представляется либо как тип data при использовании парольной защиты, либо как signedData — в случае использования электронной подписи отправителя для защиты целостности контейнера, в соответствии с разделом 5 [5].

В случае использования парольной защиты для контроля целостности поле macData должно содержать информацию об алгоритме и параметрах выработки парольного ключа в соответствии с 7.1 P 50.1.111—2016. Контроль целостности обеспечивается с использованием алгоритма HMAC_GOSTR3411_2012_512.

В случае использования электронной подписи поле macData отсутствует. Информация о сертификате отправителя и электронная подпись размещены в структуре signedData в соответствии с разделом 5 [4].

Данные в authSafe представляют собой результат кодирования списка объектов:

```
AuthenticatedSafe ::= SEQUENCE OF ContentInfo
-- Data if unencrypted
-- EncryptedData if password-encrypted
-- EnvelopedData if public key-encrypted
```

При использовании алгоритма Диффи-Хелмана для выработки ключа обмена данные представляются в виде EnvelopedData в соответствии с разделом 6 [5]. Шифрование содержимого и согласование ключей должны осуществляться в соответствии с рекомендациями разделов 6 и 7 [4].

AuthenticatedSafe может содержать объекты различного типа: ключи, сертификаты, списки отозванных сертификатов. В соответствии с [2]:

```
SafeBag ::= SEQUENCE {
    bagId          BAG-TYPE.&id ({PKCS12BagSet})
    bagValue       [0] EXPLICIT BAG-TYPE.&Type({PKCS12BagSet}@bagId),
    bagAttributes  SET OF PKCS12Attribute OPTIONAL
}
```

В случае представления данных в виде EnvelopedData секретный ключ может быть представлен в виде:

```
keyBag    BAG-TYPE ::=
    {KeyBag IDENTIFIED BY {bagtypes 1}}
```

bagValue в этом случае содержит ключ и информацию о нем в виде PrivateKeyInfo.

При использовании парольной защиты для хранения ключа должен использоваться тип:

```
pkcs8ShroudedKeyBag BAG-TYPE ::=
    {PKCS8ShroudedKeyBag IDENTIFIED BY {bagtypes 2}}
```


bagValue в этом случае содержит ключ и информацию о нем в зашифрованном виде EncryptedPrivateKeyInfo.

При шифровании сертификата с использованием парольной защиты зашифрованная структура CertBag помещается в структуру EncryptedData в соответствии с разделом 8 [5]:

```
EncryptedData ::= SEQUENCE {
    version Version,
    encryptedContentInfo EncryptedContentInfo }
```

8 Модули ASN.1

```
PKCS-12RU {iso(1) member-body(2) ru(643) rosstandart(7) tc26(1) modules(0)
    pkcs-12ruSyntax(5) }
```

```
DEFINITIONS EXPLICIT TAGS ::=
```

```
BEGIN
```

```
IMPORTS
```

```
    GostR3410-2012-PublicKey
```

```
    FROM GostR3410-2012-PKISyntax
```

```
    { iso(1) member-body(2) ru(643) rosstandart(7) tc26(1)
      modules(0) gostR3410-2012-PKISyntax(2) };
```

```
GostR3410-2012-KeyValueMask ::= OCTET STRING
```

```
GostR3410-2012-KeyValueInfo ::= SEQUENCE{
```

```
    gostR3410-2012-KeyValueMask GostR3410-2012-KeyValueMask,
```

```
    gostR3410-2012-PublicKey GostR3410-2012-PublicKey
```

```
}
```

```
GostR3410-2012-PrivateKey ::= CHOICE {
```

```
    gostR3410-2012-KeyValueMask GostR3410-2012-KeyValueMask,
```

```
    gostR3410-2012-KeyValueInfo GostR3410-2012-KeyValueInfo
```

```
}
```

```
END
```

Приложение А
(справочное)

Контрольные примеры

В данном приложении приведен пример контейнера, зашифрованного на пароле «Пароль для PFX» в соответствии с изложенной в данном документе схемой.

A1 Сертификаты

В данном разделе представлены используемые в контрольных примерах сертификаты и соответствующие им закрытые ключи.

A1.1 Корневой сертификат

Значение корневого сертификата закодированное с соответствии с алгоритмом BASE64:

```
MIICVjCCAmmgAwIBAgIQAdBoXvL8TSAAAAALJwkAATAMBggqhQMHAQEDAgUAMGAX
CzAJBgNVBAYTA1JVMRUwEwYDVoQHDAzQnNC+0YHQutCy0LAXDzANBgNVBAoMBtCi
0JoyNjEpMCCGA1UEAwggQ0EgY2VydG1maWNhdGUgKFBkLQ1MjMTIgzXhbbXBsZSku
HhcNMTUwMzI3MDcyMzAwWWhcNMjAwMzI3MDcyMzAwWjBGMQswCQYDVQGEWJSVTEV
MBMGA1UEBww0JzQvtGB0LRQstCwMQ8wDQYDVQQKDABQotCaMjYxKTAhBgNVBAMM
IENBIGN1cnRpZm1jYXR1ICChQSONTIzEyIGV4YW1wbGUwMGYwHwYIKoUDBwEBAQEw
EwYHKOUDAgIjAQYIKoUDBwEBAQIDQwAEQBxYC72z7PQ0LZCzWE1iXy7kNPks570v
ENM2iUsWGWc0pk37mkGFBUMfk13VkJamj1Czr/v/Ab49c/GcCqJap2eBCQAYnzA5
MDAwMYIjADI3MDkMDAxo4HfMIHcMA4GA1UdDwEB/wQEAwIBxjAPBgNVHRMBAf8E
BTADAQH/MB0GA1UdDgQWBQmnc7Xh5ykb5t/BMwOkxA4drfEmjCBmQYDVR0jBIGR
MIG0GgBQmnc7Xh5ykb5t/BMwOkxA4drfEmjFkpGIwYDELMAkGA1UEBhMCU1UxFTAT
BgNVBACMDNCC0L7RgdC60LLQsDEPMA0GA1UECgwGOKLQmji2MSkwJwYDVQDDCBD
QSBjZXJ0awZpY2FOZSAoUEtDUyMxMiB1eGFtcGx1KYIQAdBoXvL8TSAAAAALJwKA
ATAMBggqhQMHAQEDAgUAA0EA++0azMpEpK+nTLytJKOYmr6RoeGtfSjxfUhlfsx8
u1Jqzr9wEMK55pMNjMa8upPRiSmV8oZ+aw4ihq3Lt18hfQ==
```

Представление ASN.1 корневого сертификата:

```
0 30 702: SEQUENCE {
  4 30 617: SEQUENCE {
    8 AO 3: [0] {
      10 02 1: INTEGER 2
      :
      13 02 16: INTEGER
      : 01 D0 68 5E F2 FC 4D 20 00 00 00 0B 27 09 00 01
      31 30 12: SEQUENCE {
        33 06 8: OBJECT IDENTIFIER '1 2 643 7 1 1 3 2'
        43 05 0: NULL
        :
        45 30 96: SEQUENCE {
          47 31 11: SET {
            49 30 9: SEQUENCE {
              51 06 3: OBJECT IDENTIFIER '2 5 4 6'
              56 13 2: PrintableString 'RU'
              :
              :
            }
            60 31 21: SET {
              62 30 19: SEQUENCE {
                64 06 3: OBJECT IDENTIFIER '2 5 4 7'
                69 0C 12: UTF8String 'Москва'
                :
                :
              }
            }
            83 31 15: SET {
              85 30 13: SEQUENCE {
                87 06 3: OBJECT IDENTIFIER '2 5 4 10'
                92 0C 6: UTF8String 'TK26'
                :
                :
              }
            }
          }
        }
      }
    }
  }
}
```

```

100 31 41:      SET {
102 30 39:      SEQUENCE {
104 06 3:      OBJECT IDENTIFIER '2 5 4 3'
109 0C 32:      UTF8String 'CA certificate (PKCS#12 example)'
:      }
:      }
:      }
143 30 30:      SEQUENCE {
145 17 13:      UTCTime '150327072300Z'
160 17 13:      UTCTime '200327072300Z'
:      }
175 30 96:      SEQUENCE {
177 31 11:      SET {
179 30 9:      SEQUENCE {
181 06 3:      OBJECT IDENTIFIER '2 5 4 6'
186 13 2:      PrintableString 'RU'
:      }
:      }
190 31 21:      SET {
192 30 19:      SEQUENCE {
194 06 3:      OBJECT IDENTIFIER '2 5 4 7'
199 0C 12:      UTF8String 'Москва'
:      }
:      }
213 31 15:      SET {
215 30 13:      SEQUENCE {
217 06 3:      OBJECT IDENTIFIER '2 5 4 10'
222 0C 6:      UTF8String 'TK26'
:      }
:      }
230 31 41:      SET {
232 30 39:      SEQUENCE {
234 06 3:      OBJECT IDENTIFIER '2 5 4 3'
239 0C 32:      UTF8String 'CA certificate (PKCS#12 example)'
:      }
:      }
:      }
273 30 102:     SEQUENCE {
275 30 31:     SEQUENCE {
277 06 8:      OBJECT IDENTIFIER '1 2 643 7 1 1 1 1'
287 30 19:     SEQUENCE {
289 06 7:      OBJECT IDENTIFIER '1 2 643 2 2 35 1'
298 06 8:      OBJECT IDENTIFIER '1 2 643 7 1 1 2 2'
:      }
:      }
308 03 67:     BIT STRING 0 unused bits, encapsulates {
311 04 64:     OCTET STRING
:         1C 58 0B BD B3 EC F4 0E 2D 90 B3 58 49 62 5F 2E
:         E4 34 F9 2C E7 BD 2F 10 D3 36 89 4B 16 1B 00 B4
:         A6 4D FB 9A 41 85 05 49 9F 92 5D D5 90 96 A6 8E
:         50 B3 AF FB FF 01 BE 3D 73 F1 9C 0A A2 5A A7 67
:      }
:      }
399 A3 223:    [3] {
402 30 220:    SEQUENCE {
405 30 14:    SEQUENCE {
407 06 3:      OBJECT IDENTIFIER '2 5 29 15'
412 01 1:      BOOLEAN TRUE
415 04 4:      OCTET STRING, encapsulates {
417 03 2:      BIT STRING 1 unused bits
:          '1100011'B
:      }
:      }
:      }

```

P 50.1.112—2016

```

421 30  15:      SEQUENCE {
423 06   3:         OBJECT IDENTIFIER '2 5 29 19'
428 01   1:         BOOLEAN TRUE
431 04   5:         OCTET STRING, encapsulates {
433 30   3:           SEQUENCE {
435 01   1:             BOOLEAN TRUE
                        :
                        :           }
                        :
                        :         }
438 30  29:      SEQUENCE {
440 06   3:         OBJECT IDENTIFIER '2 5 29 14'
445 04  22:         OCTET STRING, encapsulates {
447 04  20:           OCTET STRING
                        :           26 9D CE D7 87 9C A4 6F 9B 7F 04 CC 0E 93 10 38
                        :           76 B7 C4 9A
                        :           }
                        :
                        :         }
469 30 153:      SEQUENCE {
472 06   3:         OBJECT IDENTIFIER '2 5 29 35'
477 04 145:         OCTET STRING, encapsulates {
480 30 142:           SEQUENCE {
483 80  20:             [0]
                        :             26 9D CE D7 87 9C A4 6F 9B 7F 04 CC 0E 93 10 38
                        :             76 B7 C4 9A
                        :             [1] {
                        :             [4] {
                        :               SEQUENCE {
                        :                 SET {
                        :                   SEQUENCE {
                        :                     OBJECT IDENTIFIER '2 5 4 6'
                        :                     PrintableString 'RU'
                        :                   }
                        :                 }
                        :               SET {
                        :                 SEQUENCE {
                        :                   OBJECT IDENTIFIER '2 5 4 7'
                        :                   UTF8String 'Москва'
                        :                 }
                        :               }
                        :             SET {
                        :               SEQUENCE {
                        :                 OBJECT IDENTIFIER '2 5 4 10'
                        :                 UTF8String 'TK26'
                        :               }
                        :             }
                        :           SET {
                        :             SEQUENCE {
                        :               OBJECT IDENTIFIER '2 5 4 3'
                        :               UTF8String 'CA certificate (PKCS#12 example)'
                        :             }
                        :           }
                        :         }
                        :       }
                        :     }
607 82  16:      [2]
                        :      01 D0 68 5E F2 FC 4D 20 00 00 00 0B 27 09 00 01
                        :      }
                        :    }
                        :  }
                        : }

```

```

625 30 12: SEQUENCE {
627 06 8:   OBJECT IDENTIFIER '1 2 643 7 1 1 3 2'
637 05 0:   NULL
      :   }
639 03 65: BIT STRING 0 unused bits
      :   FB E3 9A CC CA 44 A4 AF A7 4C BC AD 24 A3 98 9A
      :   BE 91 A1 E1 AD 7D 28 D7 7D 48 4B 7E CC 7C BB 52
      :   6A CE BF 70 10 C2 B9 E6 93 0D 8C C6 BC BA 93 D1
      :   89 29 95 F2 86 7E 6B 0E 22 86 AD CB B6 5F 21 7D
      :   }

```

A1.2 Тестовый сертификат

Тестовый сертификат присутствует во всех приведенных контрольных примерах в качестве сертификата, подлежащего упаковке в контейнер. Значение тестового сертификата, закодированное с соответствии с алгоритмом BASE64:

```

MIIDAjCCAq2gAwIBAgIQAdBoXzEFlsAAAAALJwkaATAMBggqhQMHAQEDAgUAMGAX
CzAJBgNVBAYTA1JVMRUwEwYDVoQQHDAzQnNC+0YHQutCyOLAxDzANBgNVBAoMBtCi
0JoyNjEpMCCGA1UEAwgQ0EgY2VydG1maWwNhdGUgKFBFLQ1MjMTIgdXhxbXBsZSkw
HhcNMTUwMzI3MDcyNTAwWhcNMjAwMzI3MDcyMzAwWjBkMQswCQYDVQQGEwJSVTEV
MBMGA1UEBwwM0JzQvTGB0LrQstCwMQ8wDQYDVQQKDABoQotCaMjYxLTArBgNVBAMM
JFRlc3QgY2VydG1maWwNhdGUgMSAoUEtDUyMxMiBlEGFtcGx1KTBmMB8GCCqFAwCB
AQEBMBMGByqFAwICAwEgCCqFAwCBQAQICAOBABEDXHPKaSm+vZ1g1PxZM5fc033r/
6Eaxc3K1RCmRYHkiYkzi2D0CwLhEhTBXkfjUyEbS4FEXB5PM3oCwB0G+FMKVgQkA
MjcwTAAwMDGjggEpMIIBJTAhBgNVHRAEJDAigA8yMDE1MDMyNzA3MjUwMFqBDzIw
MTYwMzI3MDcyNTAwWjA0BgNVHQ8BAf8EBAMCBPAwHQYDVR0OBBYEF6x6FDrzJg
3ZS2D+jAehZyqxVtMBOGA1UdJQQwMBQGCCsGAQUFBwMCGgggrBgEFBQcDBDAMBGNV
HRMBAf8EAjAAMIGZBgNVHSMGZEWgY6AFCadzteHnKRvm38EzA6TEDh2t8SaoWsk
YjBgMQswCQYDVQQGEwJSVTEVMBMGA1UEBwwM0JzQvTGB0LrQstCwMQ8wDQYDVQQK
DABoQotCaMjYxKTArBgNVBAMMIENBIGN1cnRpZm1jYXR1ICQSONTIzEyIGV4YW1w
bcUgphABOGhe8vxNIAAAAAsnCQABMAwGCCqFAwCBQAQICBAQADQD2i rRW+TySSAjC
SnTHQn14q2Jrgw10LAoCbuOCcjkjHc73wFOFpNfd1CESjZEv21MI+vrAUyF54n5h
OYxF5e+y

```

Представление ASN.1 тестового сертификата:

```

0 30 770: SEQUENCE {
4 30 685: SEQUENCE {
8 A0 3:   [0] {
10 02 1:   INTEGER 2
      :   }
13 02 16:   INTEGER
      :   01 D0 68 5F 31 1F 96 C0 00 00 00 0B 27 09 00 01
31 30 12:   SEQUENCE {
33 06 8:   OBJECT IDENTIFIER '1 2 643 7 1 1 3 2'
43 05 0:   NULL
      :   }
45 30 96:   SEQUENCE {
47 31 11:   SET {
49 30 9:     SEQUENCE {
51 06 3:       OBJECT IDENTIFIER '2 5 4 6'
56 13 2:       PrintableString 'RU'
      :       }
      :     }
60 31 21:   SET {
62 30 19:     SEQUENCE {
64 06 3:       OBJECT IDENTIFIER '2 5 4 7'
69 0C 12:       UTF8String 'Москва'
      :       }
      :     }
83 31 15:   SET {
85 30 13:     SEQUENCE {
87 06 3:       OBJECT IDENTIFIER '2 5 4 10'
92 0C 6:       UTF8String 'TK26'

```

```

:
:
100 31 41: SET {
102 30 39: SEQUENCE {
104 06 3: OBJECT IDENTIFIER '2 5 4 3'
109 0C 32: UTF8String 'CA certificate (PKCS#12 example)'
:
:
:
143 30 30: SEQUENCE {
145 17 13: UTCTime '150327072500Z'
160 17 13: UTCTime '200327072300Z'
:
:
175 30 100: SEQUENCE {
177 31 11: SET {
179 30 9: SEQUENCE {
181 06 3: OBJECT IDENTIFIER '2 5 4 6'
186 13 2: PrintableString 'RU'
:
:
:
190 31 21: SET {
192 30 19: SEQUENCE {
194 06 3: OBJECT IDENTIFIER '2 5 4 7'
199 0C 12: UTF8String 'Москва'
:
:
:
213 31 15: SET {
215 30 13: SEQUENCE {
217 06 3: OBJECT IDENTIFIER '2 5 4 10'
222 0C 6: UTF8String 'TK26'
:
:
:
230 31 45: SET {
232 30 43: SEQUENCE {
234 06 3: OBJECT IDENTIFIER '2 5 4 3'
239 0C 36: UTF8String 'Test certificate 1 (PKCS#12 example)'
:
:
:
:
277 30 102: SEQUENCE {
279 30 31: SEQUENCE {
281 06 8: OBJECT IDENTIFIER '1 2 643 7 1 1 1 1'
291 30 19: SEQUENCE {
293 06 7: OBJECT IDENTIFIER '1 2 643 2 2 35 1'
302 06 8: OBJECT IDENTIFIER '1 2 643 7 1 1 2 2'
:
:
:
:
312 03 67: BIT STRING 0 unused bits, encapsulates {
315 04 64: OCTET STRING
: D7 1C F2 9A 4A 6F AF 67 58 25 3F 16 4C E5 F7 0E
: DF 7A FF E8 46 B1 73 72 B5 44 29 91 60 79 22 62
: 4C E2 D8 3D 02 C0 B8 44 85 30 57 91 F8 D4 C8 46
: D2 E0 51 17 07 93 CC DE 80 B0 07 41 BE 14 C2 95
:
:
:
381 81 9: [1] '.27090001'
392 A3 297: [3] {
396 30 293: SEQUENCE {
400 30 43: SEQUENCE {
402 06 3: OBJECT IDENTIFIER '2 5 29 16'
407 04 36: OCTET STRING, encapsulates {
409 30 34: SEQUENCE {

```

```

411 80 15:          [0] '20150327072500Z'
428 81 15:          [1] '20160327072500Z'
      :
      :          }
      :
      :          }
445 30 14:    SEQUENCE {
447 06  3:      OBJECT IDENTIFIER '2 5 29 15'
452 01  1:      BOOLEAN TRUE
455 04  4:      OCTET STRING, encapsulates {
457 03  2:          BIT STRING 4 unused bits
      :          '1111'B
      :          }
      :      }
461 30 29:    SEQUENCE {
463 06  3:      OBJECT IDENTIFIER '2 5 29 14'
468 04 22:      OCTET STRING, encapsulates {
470 04 20:          OCTET STRING
      :          21 58 EB 11 43 AF 32 60 DD 94 B6 0F E8 C0 7A 16
      :          72 AB 15 6D
      :          }
      :      }
492 30 29:    SEQUENCE {
494 06  3:      OBJECT IDENTIFIER '2 5 29 37'
499 04 22:      OCTET STRING, encapsulates {
501 30 20:          SEQUENCE {
503 06  8:              OBJECT IDENTIFIER '1 3 6 1 5 5 7 3 2'
513 06  8:              OBJECT IDENTIFIER '1 3 6 1 5 5 7 3 4'
      :          }
      :      }
523 30 12:    SEQUENCE {
525 06  3:      OBJECT IDENTIFIER '2 5 29 19'
530 01  1:      BOOLEAN TRUE
533 04  2:      OCTET STRING, encapsulates {
535 30  0:          SEQUENCE {}
      :          }
      :      }
537 30 153:   SEQUENCE {
540 06  3:      OBJECT IDENTIFIER '2 5 29 35'
545 04 145:   OCTET STRING, encapsulates {
548 30 142:   SEQUENCE {
551 80  20:       [0]
      :       26 9D CE D7 87 9C A4 6F 9B 7F 04 CC 0E 93 10 38
      :       76 B7 C4 9A
573 A1 100:   [1] {
575 A4  98:       [4] {
577 30  96:           SEQUENCE {
579 31  11:               SET {
581 30  9:                   SEQUENCE {
583 06  3:                       OBJECT IDENTIFIER '2 5 4 6'
588 13  2:                       PrintableString 'RU'
      :                   }
      :               }
592 31 21:       SET {
594 30 19:           SEQUENCE {
596 06  3:               OBJECT IDENTIFIER '2 5 4 7'
601 0C 12:               UTF8String 'Москва'
      :           }
      :       }
615 31 15:   SET {
617 30 13:       SEQUENCE {
619 06  3:           OBJECT IDENTIFIER '2 5 4 10'

```

```

624 0C   6:           UTF8String 'TK26'
        :           }
        :           }
632 31   41:         SET {
634 30   39:         SEQUENCE {
636 06    3:           OBJECT IDENTIFIER '2 5 4 3'
641 0C   32:           UTF8String 'CA certificate (PKCS#12 example)'
        :           }
        :           }
        :           }
        :           }
        :           }
        :           }
        :           }
675 82   16:         [2]
        :           01 D0 68 5E F2 FC 4D 20 00 00 0B 27 09 00 01
        :           }
        :           }
        :           }
        :           }
        :           }
        :           }
693 30   12:         SEQUENCE {
695 06    8:           OBJECT IDENTIFIER '1 2 643 7 1 1 3 2'
705 05    0:           NULL
        :           }
707 03   65:         BIT STRING 0 unused bits
        :           F6 8A B4 56 F9 3C 92 48 08 C2 4A 74 C7 42 79 78
        :           AB 62 6B 83 0D 4E 2C 0A 02 6E E3 82 70 99 23 1D
        :           CE F7 C0 53 85 A4 D7 DD 94 21 12 8D 91 2F DA 53
        :           08 FA FA C0 53 21 79 E2 7E 61 D1 8C 45 E5 EF B2
        :           }

```

A2 Контрольный пример 1

В данном разделе приведен пример контейнера, зашифрованного на пароле «Пароль для PFX» в соответствии с изложенной в данном документе схемой применения парольной защиты. Значение контейнера, закодированное с соответствии с алгоритмом BASE64:

```

MIIFqgIBAzcCBSsGCSqGSib3DQEHAaCCBRwEggUYMIIFFDCCASIGCSqGSib3DQEH
AaCCARMEggEPMIIBCzCCAQCgGcyqGSib3DQEMcGECoiHgMIHdMHEGCSqGSib3DQEF
DTBkMEEGCSqGSib3DQEFDDA0BCD5qZr0TTIsBvdgUoq/zFwOzdyJohj6/4Wiycgg
j9AK/QICB9AwDAYIKoUDBwEBBAIFADAFBgYqhQMCAhUwFQQI3Ip/Vp0IsyIGCSqF
AwcBAGUBAQRoSfLhgX9s/zn+BjnhTORor07vS55Ys5hgvVpWDX4mXGWwyez/2sMc
aFgSr4H4UTGgwoMynGLpF1IOVo+bGJOePqHB+gS5OL9oV+PumZ/ELrRENK1CDqfY
WwpSystX29cVFrnTnDsbBYxFTATBgkqhkiG9w0BcRUxBGQEAQAADCCA+oGCSqG
SIb3DQEHbqCCA9swggPXAqEAMIID0AYJKoZiHvcNAQcBMHEGCSqGSib3DQEFDTBk
MEEGCSqGSib3DQEFDDA0BCCJTJLZQRi1wIipQHzyjXbq7+Vw2+1280C45x8ff6kMS
VAICB9AwDAYIKoUDBwEBBAIFADAFBgYqhQMCAhUwFQQIxepowwvS11MGCSqFAwCB
AgUBAYCCA06n09P/o+eDEKoSwpvlpOLks7dKmvquKzJ81nCngvLQ5fEWL1WkxwiI
rEhm53JKLD0wy4heka1Ek011Bvc51XP9gkDkmaoBpnV/TyKIY35w16ATfeGXno1M
KoA+Ktdhv4gLnz0k2SXdkUj11JwYskXue+REA0p4m2ZsoaTmvo0Damh9Jey/5Qjy
Xe58CGnyXFzX3eU86qs4WfdWdS3NzYYOk9zzV1461e9u790/LnW2j4n2of/Jpk/L
YjrRmz5oYeQoKqKohEyhP06e+ejr6laduEv7TwJQKRNIygogbVvkNn3VjHTSOUG4
W+3NRPhjB0jD9obdyx6Mwa603B9bUzFMNav8/gYn0vTDxqXmLy/92oTngNrVx6Gc
cN1128ISrDS6+RxtAMI EBRK6xNkemqX5yNXG5GrLQQFGP6mbs2nNpjk1gj3p1jmX
Eky2/G78XiJrv020gGs6CKnI9nMpa6N7PBHV34MJ6EZzWOWDRQ420xk63mni crs0
WDVJ0xjdu4FW3iEk02EaiRTvGPa6GL7LBp6Q1aXSSwONx725cyRsL9cTlukqXER
WHD1MpjYlBkGZRrCc1myWgEfspuTfSIPNF/oLv9kJNwacP3u0Dfecg3us7eg20A
xo5zrYfn39GCBMF1WHAYRO/+PnJb9jrDuLAE8+ONNqjNu11MK9CS tEhb6Te+yE6q
oeP6hJjFLi+nFLE9ymIo0A7gLQD5vzFv1+7v1ZNVnQkwRUsWoRiEvvGnv3Z1iZU6
xStxgoHM162V/P5cz4dr9vJM2adEWNZcVX16mk1H8DRc1sRGnvs21237oKWRVntJ
h0wnZ8qtD+3ZUqsX79QhVzUQBzKuBt6jwNhaHLG15B+Or/zA9Fezs0h6+Uc+fZaV
W7FFfelyWwGy90XD3ybTrjzep9f3nt55Z2c+fu2iEwhoyImwLuC3+CVhf9Af59j9
8/BophMJuATDJEtgi8rt4vLnfxKu250Mv2ZpbfF69EGTgFYbwc55zRfaUG9z1yCu

```


1YwMJ6HC9FUVtJp9g0bSsriRbzTH7mVaMjQkBLotazWbegzI+be8V3yT06C+ehD+2
 GdLWAVs9hp8gPHEUShb/XrgPpDSJmF10iye0FBO/j4edDACKqVcwdjBOMAOGCCqF
 AwcBAQIDBEAIFX0fyZe20QKKhWm6WYX+S92Gt6zaXroX0vAmayzLFZ5Sd9C2t9zZ
 JSg6M8RBUYpw/8ym5ou1o2nDa09M5zF3BCCpzyCQBI+rzfISeKvPV1R0fcXiYU93
 mwc11xQV2G5/fgICB9A=

Представление ASN.1:

```

0 30 1450: SEQUENCE {
  4 02 1: INTEGER 3
  7 30 1323: SEQUENCE {
11 06 9: OBJECT IDENTIFIER '1 2 840 113549 1 7 1'
22 A0 1308: [0] {
26 04 1304: OCTET STRING, encapsulates {
30 30 1300: SEQUENCE {
34 30 290: SEQUENCE {
38 06 9: OBJECT IDENTIFIER '1 2 840 113549 1 7 1'
49 A0 275: [0] {
53 04 271: OCTET STRING, encapsulates {
57 30 267: SEQUENCE {
61 30 263: SEQUENCE {
65 06 11: OBJECT IDENTIFIER '1 2 840 113549 1 12 10 1 2'
78 A0 224: [0] {
81 30 221: SEQUENCE {
84 30 113: SEQUENCE {
86 06 9: OBJECT IDENTIFIER '1 2 840 113549 1 5 13'
97 30 100: SEQUENCE {
99 30 65: SEQUENCE {
101 06 9: OBJECT IDENTIFIER '1 2 840 113549 1 5 12'
112 30 52: SEQUENCE {
114 04 32: OCTET STRING
: F9 A9 9A F4 4D 32 2C 06 F7 60 52 8A BF CC 5C 0E
: CD DC 89 A2 18 FA FF 85 A2 C9 C7 20 8F D0 0A FD
148 02 2: INTEGER 2000
152 30 12: SEQUENCE {
154 06 8: OBJECT IDENTIFIER '1 2 643 7 1 1 4 2'
164 05 0: NULL
: }
: }
166 30 31: SEQUENCE {
168 06 6: OBJECT IDENTIFIER '1 2 643 2 2 21'
176 30 21: SEQUENCE {
178 04 8: OCTET STRING
: DC 8A 7F 56 9D 08 B3 22
188 06 9: OBJECT IDENTIFIER '1 2 643 7 1 2 5 1 1'
: }
: }
: }
199 04 104: OCTET STRING
: 49 F2 E1 83 1F 6C FF 39 FE 06 39 E1 4F 4A E8 AF
: 4E EF 4B 9E 58 B3 98 60 BD 5A 56 0F 1E 26 5C 65
: 96 C9 EC FF DA C3 1C 68 58 12 AF 81 F8 51 31 86
: C2 83 32 9C 62 E9 17 52 0E 56 8F 9B 18 9D 1E 3E
: A1 C1 FA 04 B9 38 BF 68 57 E3 D4 99 9F C4 2E B4
: 44 34 A9 42 0E A7 D8 5A FA 52 CA CB 57 DB D0 AF
: 08 5A E7 4E 70 EC 6C 16
: }
: }
305 31 21: SET {
307 30 19: SEQUENCE {
309 06 9: OBJECT IDENTIFIER '1 2 840 113549 1 9 21'

```

P 50.1.112—2016

```
320 31 6:          SET {
322 04 4:          OCTET STRING
:              01 00 00 00
:              }
:              }
:              }
:              }
:              }
:              }
:              }
:              }
:          }
328 30 1002:      SEQUENCE {
332 06 9:          OBJECT IDENTIFIER '1 2 840 113549 1 7 6'
343 A0 987:      [0] {
347 30 983:          SEQUENCE {
351 02 1:          INTEGER 0
354 30 976:          SEQUENCE {
358 06 9:          OBJECT IDENTIFIER '1 2 840 113549 1 7 1'
369 30 113:          SEQUENCE {
371 06 9:          OBJECT IDENTIFIER '1 2 840 113549 1 5 13'
382 30 100:          SEQUENCE {
384 30 65:          SEQUENCE {
386 06 9:          OBJECT IDENTIFIER '1 2 840 113549 1 5 12'
397 30 52:          SEQUENCE {
399 04 32:          OCTET STRING
:              89 4C 92 D9 41 18 B5 58 8A 50 1F 3C A3 5D BA BB
:              F9 5C 36 FB 5D BC D0 2E 39 C7 C7 DF EA 43 12 54
433 02 2:          INTEGER 2000
437 30 12:          SEQUENCE {
439 06 8:          OBJECT IDENTIFIER '1 2 643 7 1 1 4 2'
449 05 0:          NULL
:          }
:      }
:      }
451 30 31:          SEQUENCE {
453 06 6:          OBJECT IDENTIFIER '1 2 643 2 2 21'
461 30 21:          SEQUENCE {
463 04 8:          OCTET STRING
:          C5 EA 68 C3 0B D2 D7 53
473 06 9:          OBJECT IDENTIFIER '1 2 643 7 1 2 5 1 1'
:          }
:      }
:      }
:      }
484 80 846:      [0]
:          A7 D3 D3 FF A3 E7 83 10 AA 12 5A 9B E5 A4 E2 CA
:          B3 B7 4A 99 5A AE 2B 32 7C D6 70 A7 82 F2 D0 E5
:          F1 16 2F 55 A4 C7 08 88 AC 48 66 E7 72 4A 2C 3D
:          30 CB 88 5E 91 A9 44 93 4D 75 06 F7 39 D5 73 FD
:          82 40 E4 99 AA 01 A6 75 7F 4F 22 88 63 7E 70 97
:          A0 13 7D E1 97 9E 8D 4C 2A 80 3E 2A D7 61 BF 88
:          0B 9F 3D 24 D9 25 DD 91 48 F5 D4 9C 18 B2 45 EE
:          7B E4 44 03 4A 78 9B 66 6C A1 A4 E6 BE 83 83 6A
:          [ Another 718 bytes skipped ]
:      }
:      }
:      }
:      }
:      }
:      }
:      }
:      }
```

```

1334 30 118: SEQUENCE {
1336 30 78:   SEQUENCE {
1338 30 10:     SEQUENCE {
1340 06 8:       OBJECT IDENTIFIER '1 2 643 7 1 1 2 3'
:         }
1350 04 64:       OCTET STRING
:         08 15 7D 1F C9 97 B6 D1 02 8A 85 69 BA 59 85 FE
:         4B DD 86 B7 AC DA 5E BA 17 3A F0 26 6B 2C CB 7D
:         9E 52 77 D0 B6 B7 DC D9 25 28 3A 33 C4 41 51 8A
:         70 FF CC A6 E6 8B B5 A3 69 C3 6B 4F 4C E7 31 77
:       }
1416 04 32:       OCTET STRING
:         A9 CF 20 90 04 8F AB CD F2 12 78 AB CF 57 54 4E
:         7D C5 E2 61 4F 77 9B 07 25 D7 14 15 D8 6E 7F 7E
1450 02 2:       INTEGER 2000
:     }
:   }

```

Представление пароля в бинарном виде:

```
d0 9f d0 b0 d1 80 d0 be d0 bb d1 8c 20 d0 b4 d0 bb d1 8f 20 50 46 58
```

Значение ключа шифрования закрытого ключа:

```
30 9d d0 35 4c 56 03 73 94 03 f2 33 5e 9e 20 55
13 8f 8b 5c 98 b6 30 09 de 06 35 ee a1 fd 7b a8
```

Синхроросылка:

```
dc 8a 7f 56 9d 08 b3 22
```

Значение расшифрованного закрытого ключа, закодированное в соответствии с алгоритмом BASE64:

```
MGYCAQAwHwYIKoUDBwEBAQEwEwYHkoUDAgIjAQYIKoUDBwEBAgIEQEYbRu86z+1JFKDcPDN9UbTG
G2ki9enTqos4KpUU0j9IDp11UXiaA1YDIwUj1Ap+81GkLmyt8Fw6Gt/X5JZySAY=
```

Представление ASN.1 закрытого ключа:

```

0 30 102: SEQUENCE {
2 02 1:   INTEGER 0
5 30 31:  SEQUENCE {
7 06 8:   OBJECT IDENTIFIER '1 2 643 7 1 1 1 1'
17 30 19:  SEQUENCE {
19 06 7:   OBJECT IDENTIFIER '1 2 643 2 2 35 1'
28 06 8:   OBJECT IDENTIFIER '1 2 643 7 1 1 2 2'
:     }
:   }
38 04 64:  OCTET STRING
:     46 1B 46 EF 3A CF ED 49 14 A0 DC 3C 33 7D 51 B4
:     C6 1B 69 22 F5 E9 D3 AA 8B 38 2A 95 14 D2 3F 48
:     0E 99 75 51 78 9A 03 56 03 23 05 23 94 0A 7E F3
:     51 A4 2E 6C AD F0 5C 3A 1A DF D7 E4 96 72 48 06
:   }

```

Значение ключа шифрования набора сертификатов:

```
0e 93 d7 13 39 e7 f5 3b 79 a0 bc 41 f9 10 9d d4
fb 60 b3 0a e1 07 36 c1 bb 77 b8 4c 07 68 1c fc
```

Синхроросылка:

```
c5 ea 68 c3 0b d2 d7 53
```

Значение расшифрованного текста, закодированное в соответствии с алгоритмом BASE64:

```
MIIDSjCCA0YGCyqGSIB3DQEMCGEDoIIDHjCCAxoGCiqGSIB3DQEJFgGgggMKBIIDbjCCAwIwggKt
oAMCAQICEAHQaF8xH5bAAAAACycJAAEwDAYIKoUDBwEBAwIFADBgMQswCQYDVQQGEwJSVTEVMBMG
A1UEBwwM0JzQvtGB0LrQstCwMQ8wDQYDVQQKDABoTcaMjYxKTAnBgNVBAMMIENBIGN1cnRpZm1j
YXR1ICChQSONTIzEzYIGV4Yw1wbGUyPMB4XDTE1MDMyNzA3MjUwMFoXDTIwMDMyNzA3MjUwMFowZDEL
```

```

MAKGA1UEBhMCU1UxFTATBgNVBACMDNCC0L7RgdC60LLQsDEPMA0GA1UECgwGOKLQmjiI2MS0wKwYD
VQDDCRUZXN0IGNlcnRpZm1lYXRlIDEGKFBLQ1MjMTIgzXhbbXBsZSskwZjAfBggqhQMHAQEBAAT
BgcqhQMAiMBBggqhQMHAQECAGNDAARA1xzymkpv2dYJT8WTOX3Dt96/+hGsXNytUQpkWB5ImJM
4tg9AsC4RIUw5H41MhG0uBRFweTzN6AsAdBvhtClYEJADI3MDkwMDAoxo4IBKTCASUwKwYDVROQ
BCQwIoAPmjAxNTAzMjcwNzI1MDBagQ8yMDE2MDMyNzA3MjUwMFowDgYDVROPAQH/BAQDAgTwMBOG
A1UdDgQWBqhwOsRQ68yYN2Utg/owHoWcqsVbTAdBgNVHSUEFjAUBgggRBgEFAQcDAGYIKwYBBQUH
AwQwDAYDVROTAQH/BAIwADCbmQYDVROjBGRMIG0gBQmnc7Xh5ykb5t/BMw0kxA4drfEmqFkpGIw
YDELMAKGA1UEBhMCU1UxFTATBgNVBACMDNCC0L7RgdC60LLQsDEPMA0GA1UECgwGOKLQmjiI2MSkw
JwYDVQDDCDBDQSBjZXJOaWZpY2F0ZSAoUeTdUyMxMiBlEGFtcGx1KYIQAdBoXvL8TSAAAAALJwKA
ATAMBggqhQMHAQEDAGUAA0EA9oq0Vvk8kkkGIwkw0x0J5eKtia4MNTiwKAj7jgnCZIx3098BThaTX
3ZQhEo2RL9pTcPr6wFMheeJ+YdGMRexvsjEVMBMGCSqGSib3DQEJFTEGBAQBAAAA

```

Представление ASN.1 расшифрованного текста:

```

0 30 842: SEQUENCE {
  4 30 838: SEQUENCE {
    8 06 11: OBJECT IDENTIFIER
      : pkcs-12-certBag (1 2 840 113549 1 12 10 1 3)
  21 A0 798: [0] {
    25 30 794: SEQUENCE {
      29 06 10: OBJECT IDENTIFIER
        : x509Certificate (for PKCS #12) (1 2 840 113549 1 9 22 1)
    41 A0 778: [0] {
      45 04 774: OCTET STRING, encapsulates {
        49 30 770: SEQUENCE {
          53 30 685: SEQUENCE {
            57 A0 3: [0] {
              59 02 1: INTEGER 2
            }
          62 02 16: INTEGER
            : 01 D0 68 5F 31 1F 96 C0 00 00 00 0B 27 09 00 01
          80 30 12: SEQUENCE {
            82 06 8: OBJECT IDENTIFIER '1 2 643 7 1 1 3 2'
            92 05 0: NULL
          }
        94 30 96: SEQUENCE {
          96 31 11: SET {
            98 30 9: SEQUENCE {
              100 06 3: OBJECT IDENTIFIER countryName (2 5 4 6)
              105 13 2: PrintableString 'RU'
            }
          109 31 21: SET {
            111 30 19: SEQUENCE {
              113 06 3: OBJECT IDENTIFIER localityName (2 5 4 7)
              118 0C 12: UTF8String 'Москва'
            }
          132 31 15: SET {
            134 30 13: SEQUENCE {
              136 06 3: OBJECT IDENTIFIER
                : organizationName (2 5 4 10)
              141 0C 6: UTF8String 'TK26'
            }
          149 31 41: SET {
            151 30 39: SEQUENCE {
              153 06 3: OBJECT IDENTIFIER commonName (2 5 4 3)
              158 0C 32: UTF8String 'CA certificate (PKCS#12 example)'
            }
          }
        192 30 30: SEQUENCE {
          194 17 13: UTCTime '150327072500Z'
        }
      }
    }
  }
}

```

```

209 17 13:      UTCTime '200327072300Z'
                }
224 30 100:    SEQUENCE {
226 31 11:     SET {
228 30 9:      SEQUENCE {
230 06 3:      OBJECT IDENTIFIER countryName (2 5 4 6)
235 13 2:      PrintableString 'RU'
                }
                }
239 31 21:    SET {
241 30 19:     SEQUENCE {
243 06 3:      OBJECT IDENTIFIER localityName (2 5 4 7)
248 0C 12:     UTF8String ' Москва'
                }
                }
262 31 15:    SET {
264 30 13:     SEQUENCE {
266 06 3:      OBJECT IDENTIFIER
                organizationName (2 5 4 10)
271 0C 6:      UTF8String 'TK26'
                }
                }
279 31 45:    SET {
281 30 43:     SEQUENCE {
283 06 3:      OBJECT IDENTIFIER commonName (2 5 4 3)
288 0C 36:     UTF8String 'Test certificate 1 (PKCS#12 example)'
                }
                }
326 30 102:   SEQUENCE {
328 30 31:     SEQUENCE {
330 06 8:      OBJECT IDENTIFIER '1 2 643 7 1 1 1 1'
340 30 19:     SEQUENCE {
342 06 7:      OBJECT IDENTIFIER '1 2 643 2 2 35 1'
351 06 8:      OBJECT IDENTIFIER '1 2 643 7 1 1 2 2'
                }
                }
361 03 67:    BIT STRING 0 unused bits, encapsulates {
364 04 64:    OCTET STRING
                D7 1C F2 9A 4A 6F AF 67 58 25 3F 16 4C E5 F7 0E
                DF 7A FF E8 46 B1 73 72 B5 44 29 91 60 79 22 62
                4C E2 D8 3D 02 C0 B8 44 85 30 57 91 F8 D4 C8 46
                D2 E0 51 17 07 93 CC DE 80 B0 07 41 BE 14 C2 95
                }
430 81 9:     [1] '.27090001'
441 A3 297:   [3] {
445 30 293:   SEQUENCE {
449 30 43:     SEQUENCE {
451 06 3:      OBJECT IDENTIFIER
                privateKeyUsagePeriod (2 5 29 16)
456 04 36:    OCTET STRING, encapsulates {
458 30 34:     SEQUENCE {
460 80 15:     [0] '20150327072500Z'
477 81 15:     [1] '20160327072500Z'
                }
                }
494 30 14:    SEQUENCE {
496 06 3:      OBJECT IDENTIFIER keyUsage (2 5 29 15)
501 01 1:      BOOLEAN TRUE
504 04 4:      OCTET STRING, encapsulates {
506 03 2:      BIT STRING 4 unused bits

```

```

:           '1111'B
:           }
:         }
510 30 29: SEQUENCE {
512 06 3:   OBJECT IDENTIFIER
:         subjectKeyIdentifier (2 5 29 14)
517 04 22:   OCTET STRING, encapsulates {
519 04 20:   OCTET STRING
:         21 58 EB 11 43 AF 32 60 DD 94 B6 0F E8 C0 7A 16
:         72 AB 15 6D
:         }
:       }
541 30 29: SEQUENCE {
543 06 3:   OBJECT IDENTIFIER extKeyUsage (2 5 29 37)
548 04 22:   OCTET STRING, encapsulates {
550 30 20:   SEQUENCE {
552 06 8:   OBJECT IDENTIFIER
:         clientAuth (1 3 6 1 5 5 7 3 2)
562 06 8:   OBJECT IDENTIFIER
:         emailProtection (1 3 6 1 5 5 7 3 4)
:       }
:     }
572 30 12: SEQUENCE {
574 06 3:   OBJECT IDENTIFIER
:         basicConstraints (2 5 29 19)
579 01 1:   BOOLEAN TRUE
582 04 2:   OCTET STRING, encapsulates {
584 30 0:   SEQUENCE {}
:     }
:   }
586 30 153: SEQUENCE {
589 06 3:   OBJECT IDENTIFIER
:         authorityKeyIdentifier (2 5 29 35)
594 04 145:   OCTET STRING, encapsulates {
597 30 142:   SEQUENCE {
600 80 20:   [0]
:         26 9D CE D7 87 9C A4 6F 9B 7F 04 CC 0E 93 10 38
:         76 B7 C4 9A
:       [1] {
:         [4] {
:         SEQUENCE {
:         SET {
:         SEQUENCE {
:         OBJECT IDENTIFIER
:         countryName (2 5 4 6)
:         PrintableString 'RU'
:       }
:     }
:   SET {
:   SEQUENCE {
:   OBJECT IDENTIFIER
:   localityName (2 5 4 7)
:   UTF8String 'Москва'
: }
: }
: SET {
: SEQUENCE {
: OBJECT IDENTIFIER
: organizationName (2 5 4 10)
: UTF8String 'TK26'
: }
: }
: }

```


Библиография

- [1] PKCS#8 PKCS#8 (версия 1.2) Стандарт на синтаксис информации закрытого ключа [Private-Key Information Syntax Standard (v. 1.2), RSA Laboratories, 1993]
- [2] PKCS#12 PKCS#12 (версия 1.0) Синтаксис обмена персональной информацией [Personal Information Exchange Syntax (v. 1.0), RSA Laboratories, 1999]
- [3] Методические рекомендации ТК 26 Идентификаторы объектов технического комитета по стандартизации «Криптографическая защита информации» (TK26OID)
- [4] Методические рекомендации ТК 26 Информационная технология. Криптографическая защита информации. Использование алгоритмов ГОСТ 28147—89, ГОСТ Р 34.11 и ГОСТ Р 34.10 в криптографических сообщениях формата CMS (TK26CMS)
- [5] RFC5652 Р. Хаусли. Синтаксис криптографических сообщений [R. Housley. Cryptographic Message Syntax (CMS), Standards Track, IETF RFC5652, September 2009]

УДК 681.3.06:006.354

ОКС 35. 040

ОКСТУ 5002

П85

Ключевые слова: криптографические протоколы, транспортный ключевой контейнер, аутентификация, пароль, ключ

Редактор *И.А. Сериков*
Технический редактор *В.Н. Прусакова*
Корректор *И.А. Королева*
Компьютерная верстка *А.С. Тыртышного*

Сдано в набор 28.11.2016. Подписано в печать 05.12.2016. Формат 60 × 84 ¹/₈. Гарнитура Ариал.
Усл. печ. л. 2,79. Уч.-изд. л. 2,52. Тираж 33 экз. Зак. 3016.

Подготовлено на основе электронной версии, предоставленной разработчиком стандарта

Издано и отпечатано во ФГУП «СТАНДАРТИНФОРМ», 123995 Москва, Гранатный пер., 4.
www.gostinfo.ru info@gostinfo.ru