

# Объекты ключей

Определение CKO\_ отсутствует для базового класса и существует только для типов ключей.

В этом разделе определяется класс объектов CKO\_PUBLIC\_KEY, CKO\_PRIVATE\_KEY и CKO\_SECRET\_KEY типа данных CK\_OBJECT\_CLASS атрибута CKA\_CLASS.

- Общие атрибуты объектов ключей
- Объекты открытых ключей
  - Объект CKO\_PUBLIC\_KEY, тип CKK\_RSA (определен стандартом)
  - Объект CKO\_PUBLIC\_KEY, тип CKK\_ECDSA (определен стандартом)
  - Объект CKO\_PUBLIC\_KEY, тип CKK\_GOSTR3410 (определен стандартом)
  - Объект CKO\_PUBLIC\_KEY, тип CKK\_GOSTR3410\_512 (определен расширением стандарта)
- Объекты закрытых ключей
  - Объект CKO\_PRIVATE\_KEY, тип CKK\_RSA (определен стандартом)
  - Объект CKO\_PRIVATE\_KEY, тип CKK\_ECDSA (определен стандартом)
  - Объект CKO\_PRIVATE\_KEY, тип CKK\_GOSTR3410 (определен стандартом)
  - Объект CKO\_PRIVATE\_KEY, тип CKK\_GOSTR3410\_512 (определен расширением стандарта)
- Объекты секретных ключей
  - Объект CKO\_SECRET\_KEY, тип CKK\_GENERIC\_SECRET (определен стандартом)
  - Объект CKO\_SECRET\_KEY, тип CKK\_GOST (определен производителем)
  - Объект CKO\_SECRET\_KEY, тип CKK\_GOST28147 (определен стандартом)

## Общие атрибуты объектов ключей

Объекты ключей содержат ключи, использующиеся для шифрования или аутентификации, и которые могут быть открытыми, закрытыми или секретными.

Следующая таблица определяет атрибуты, общие для открытых, закрытых и секретных ключей, в дополнение к общим атрибутам, определенным для этого класса объектов

Общие атрибуты объектов ключей

| Атрибут                                | Тип данных                                                   | Значение                                                                                                                                                                                                          |
|----------------------------------------|--------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| CKA_KEY_TYPE <sup>1,2</sup>            | CK_KEY_TYPE                                                  | Тип ключа                                                                                                                                                                                                         |
| CKA_ID <sup>3</sup>                    | Byte array                                                   | Идентификатор ключа (по умолчанию пусто)                                                                                                                                                                          |
| CKA_START_DATE <sup>3</sup>            | CK_DATE                                                      | Дата начала действия ключа (по умолчанию пусто)                                                                                                                                                                   |
| CKA_END_DATE <sup>3</sup>              | CK_DATE                                                      | Дата окончания действия ключа (по умолчанию пусто)                                                                                                                                                                |
| CKA_DERIVE <sup>3</sup>                | CK_BBOOL                                                     | CK_TRUE, если ключ поддерживает выработку ключей обмена (из исходного ключа можно извлечь другие ключи). По умолчанию принимает значение CK_FALSE                                                                 |
| CKA_LOCAL <sup>4,5,6</sup>             | CK_BBOOL                                                     | CK_TRUE, если ключ был сгенерирован на токене вызовом функций C_GenerateKey или C_GenerateKeyPair или создан вызовом функции C_CopyObject как копия ключа с атрибутом CKA_LOCAL, установленным в значение CK_TRUE |
| CKA_KEY_GEN_MECHANISM <sup>4,5,6</sup> | CK_MECHANISM_TYPE                                            | Идентификатор механизма, используемого при генерации ключа                                                                                                                                                        |
| CKA_ALLOWED_MECHANISMS                 | CK_MECHANISM_TYPE_PTR, указатель на массив CK_MECHANISM_TYPE | Список механизмов, которые можно использовать с этим ключом. Число механизмов в массиве равно отношению переменной ulValueLen атрибута к размеру CK_MECHANISM_TYPE.                                               |

<sup>1</sup> должен быть определен при создании объекта с помощью функции C\_CreateObject.

<sup>2</sup> должен быть определен при расшифровании объекта с помощью функции C\_UnwrapKey.

- <sup>3</sup> может быть изменен после создания объекта вызовом функции **C\_SetAttributeValue** или при копировании объекта вызовом функции **C\_CopyObject**
- <sup>4</sup> должен остаться *незаданным* при создании объекта с помощью функции **C\_CreateObject**
- <sup>5</sup> должен остаться *незаданным* при генерации объекта с помощью функций **C\_GenerateKey** или **C\_GenerateKeyPair**
- <sup>6</sup> должен остаться *незаданным* при расшифровании объекта с помощью функции **C\_UnwrapKey**

Поле **СКА\_ID** предназначено для различения ключей. В случае с открытым и закрытым ключом оно помогает управлять несколькими ключами одного и того же владельца, так как открытый ключ имеет точно такой же идентификатор, что и открытый. Также идентификатор ключа должен совпадать с идентификатором соответствующего сертификата, если он существует, однако стандарт не настаивает на выполнении этого требования.

Для секретного ключа значение атрибута **СКА\_ID** определяется приложением.

Атрибуты **СКА\_START\_DATE** и **СКА\_END\_DATE** предназначены исключительно для справки и никакого специального назначения для них не определено. В частности, ограничение использования ключа с этими датами осуществляется приложением.

Атрибут **СКА\_DERIVE** имеет значение CK\_TRUE в том и только в том случае, если из этого ключа могут быть получены другие ключи.

Атрибут **СКА\_LOCAL** имеет значение CK\_TRUE в том и только в том случае, если ключ был изначально сгенерирован на токен вызовом функции **C\_GenerateKey** или **C\_GenerateKeyPair**.

Атрибут **СКА\_KEY\_GEN\_MECHANISM** идентифицирует механизм генерации ключа с использованием данных ключа. Он содержит действующее значение, только если атрибут **СКА\_LOCAL** имеет значение CK\_TRUE. Если **СКА\_LOCAL** имеет значение CK\_FALSE, значение атрибута будет CK\_UNAVAILABLE\_INFORMATION.

[к содержанию ↑](#)

## Объекты открытых ключей

- [Объект СКО\\_PUBLIC\\_KEY, тип CKK\\_RSA \(определен стандартом\)](#)
- [Объект СКО\\_PUBLIC\\_KEY, тип CKK\\_ECDSA \(определен стандартом\)](#)
- [Объект СКО\\_PUBLIC\\_KEY, тип CKK\\_GOSTR3410 \(определен стандартом\)](#)
- [Объект СКО\\_PUBLIC\\_KEY, тип CKK\\_GOSTR3410\\_512 \(определен расширением стандарта\)](#)

Объекты открытых ключей (класс объектов **СКО\_PUBLIC\_KEY**) содержат открытые ключи. Следующая таблица определяет общие стандартные и определенные производителем атрибуты для всех открытых ключей, в дополнение к [общим атрибутам](#), определенным для этого класса объектов.

### Общие атрибуты объектов открытых ключей (стандартные и определенные производителем)

| Атрибут                                                       | Тип данных | Значение                                                                                                                              |
|---------------------------------------------------------------|------------|---------------------------------------------------------------------------------------------------------------------------------------|
| Общие атрибуты открытых ключей (Common Public Key Attributes) |            |                                                                                                                                       |
| СКА_SUBJECT <sup>1</sup>                                      | Byte array | Имя ключа в DER-кодировке (по умолчанию пусто)                                                                                        |
| СКА_ENCRYPT <sup>1</sup>                                      | CK_BBOOL   | CK_TRUE, если ключ поддерживает шифрование <sup>2</sup>                                                                               |
| СКА_VERIFY <sup>1</sup>                                       | CK_BBOOL   | CK_TRUE, если ключ поддерживает проверку подписи, представленной в виде приложения к данным <sup>2</sup>                              |
| СКА_VERIFY_RECOVER <sup>1</sup>                               | CK_BBOOL   | CK_TRUE, если ключ поддерживает проверку подписи с восстановлением (восстанавливаемую из данных) <sup>2</sup>                         |
| СКА_WRAP <sup>1</sup>                                         | CK_BBOOL   | CK_TRUE, если ключ поддерживает маскирование других ключей (то есть может быть использован для шифрования других ключей) <sup>2</sup> |

|                                                                                                              |                  |                                                                                                                                                                                                                                                           |
|--------------------------------------------------------------------------------------------------------------|------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| CKA_TRUSTED <sup>3</sup>                                                                                     | CK_BBOOL         | Ключ может быть доверенным для приложения, которым был создан. Ключ шифрования может быть использован для шифрования (маскирования) ключей со значением атрибута CKA_WRAP_WITH_TRUSTED равным CK_TRUE.                                                    |
| CKA_WRAP_TEMPLATE                                                                                            | CK_ATTRIBUTE_PTR | Для ключей шифрования ключей (маскирования). Шаблон атрибутов для всех ключей, которые могут быть зашифрованы с помощью данного ключа шифрования. Число атрибутов в массиве равно отношению переменной <i>ulValueLen</i> атрибута к размеру CK_ATTRIBUTE. |
| <b>Атрибуты открытых ключей, определенные производителем (Rutoken Vendors Defined Public Key Attributes)</b> |                  |                                                                                                                                                                                                                                                           |
| CKA_CAPI_ID                                                                                                  |                  | Идентификатор ключевой пары (по умолчанию 0)                                                                                                                                                                                                              |
| CKA_PUBLIC_KEY_RSF_ID                                                                                        |                  | Идентификатор RSF-файла, хранящего открытый ключ (по умолчанию 0)                                                                                                                                                                                         |
| CKA_PRIVATE_KEY_RSF_ID                                                                                       |                  | Идентификатор RSF-файла, хранящего соответствующий закрытый ключ (по умолчанию 0)                                                                                                                                                                         |
| CKA_VENDOR_KEY_JOURNAL                                                                                       | CK_BBOOL         | Если CK_TRUE, то ключ может использоваться только для работы с журналом РутOKEN PINPad                                                                                                                                                                    |

<sup>1</sup> может быть изменен после создания объекта с помощью функции **C\_SetAttributeValue**

<sup>2</sup> значение по умолчанию определяется токеном и может зависеть от значения других атрибутов

<sup>3</sup> значение CK\_TRUE может быть задано только Администратором

Для совместимости значения полей **CKA\_SUBJECT** и **CKA\_ID** открытого ключа должны совпадать с полями **CKA\_SUBJECT** и **CKA\_ID** соответствующих сертификата и закрытого ключа. Однако стандарт не настаивает на выполнении этого требования, так же как и не является обязательным хранение сертификата и закрытого ключа на токене.

Атрибут **CKA\_CAPI\_ID** предназначен для хранения ID контейнера из библиотеки rtCSP. В случае, когда нет возможности выяснить значения атрибутов **CKA\_CAPI\_ID**, **CKA\_PUBLIC\_KEY\_RSF\_ID** и **CKA\_PRIVATE\_KEY\_RSF\_ID**, следует использовать значение по умолчанию, равное 0.

Атрибут **CKA\_VENDOR\_KEY\_JOURNAL** применим только для РутOKEN PINPad.

Следующая таблица показывает соответствие между флагами **keyUsage** стандарта ISO/IEC 9594-8 (X.509) для открытых ключей и атрибутами стандарта PKCS #11 для открытых ключей.

#### Соответствие флагов стандарта X.509 атрибутам открытых ключей стандарта PKCS #11

| Использование флагов для открытых ключей в сертификатах X.509 открытого ключа | Соответствующие атрибуты стандарта PKCS #11 для открытых ключей |
|-------------------------------------------------------------------------------|-----------------------------------------------------------------|
| dataEncipherment                                                              | CKA_ENCRYPT                                                     |
| digitalSignature, keyCertSign, cRLSign                                        | CKA_VERIFY                                                      |
| digitalSignature, keyCertSign, cRLSign                                        | CKA_VERIFY_RECOVER                                              |
| keyAgreement                                                                  | CKA_DERIVE                                                      |
| keyEncipherment                                                               | CKA_WRAP                                                        |
| nonRepudiation                                                                | CKA_VERIFY                                                      |
| nonRepudiation                                                                | CKA_VERIFY_RECOVER                                              |

[к содержанию ↑](#)

## Объект CKO\_PUBLIC\_KEY, тип CKK\_RSA (определен стандартом)

Объекты открытого ключа RSA (объект CKO\_PUBLIC\_KEY, тип CKK\_GENERIC\_SECRET) содержат открытые ключи RSA.

Объект CKO\_PUBLIC\_KEY типа CKK\_RSA содержит как общие атрибуты объектов класса **CKO\_PUBLIC\_KEY**, так и специфические, которые приведены в следующей таблице.

Атрибуты объекта CKO\_PUBLIC\_KEY, тип CKK\_RSA

| Атрибут                                                  | Тип данных  | Значение                      |
|----------------------------------------------------------|-------------|-------------------------------|
| Атрибуты открытого ключа RSA (RSA Public Key Attributes) |             |                               |
| CKA_MODULUS <sup>1,2</sup>                               | Big integer | Модуль <i>n</i>               |
| CKA_MODULUS_BITS <sup>3,4</sup>                          | CK_ULONG    | Длина модуля <i>n</i> в битах |
| CKA_PUBLIC_EXPONENT <sup>1</sup>                         | Big integer | Открытая экспонента <i>e</i>  |

- <sup>1</sup> должен быть определен при создании объекта с помощью функции **C\_CreateObject**.
- <sup>2</sup> должен остаться *незаданным* при генерации объекта с помощью функций **C\_GenerateKey** или **C\_GenerateKeyPair**
- <sup>3</sup> должен остаться *незаданным* при создании объекта с помощью функции **C\_CreateObject**
- <sup>4</sup> должен быть определен при генерации объекта с помощью функций **C\_GenerateKey** или **C\_GenerateKeyPair**

Шаблон создания открытого ключа RSA

```
CK_OBJECT_CLASS your_class = CKO_PUBLIC_KEY;
CK_KEY_TYPE keyType = CKK_GENERIC_RSA;
CK_UTF8CHAR label[] = "An RSA public_key object";
CK_BYTE modulus[] = {...};
CK_BYTE exponent[] = {...};
CK_BBOOL IsTrue = CK_TRUE;

CK_ATTRIBUTE template[] = {
    {CKA_CLASS, &your_class, sizeof(your_class)},
    {CKA_KEY_TYPE, &keyType, sizeof(keyType)},
    {CKA_TOKEN, &IsTrue, sizeof(IsTrue)},
    {CKA_LABEL, label, sizeof(label)-1},
    {CKA_WRAP, &IsTrue, sizeof(IsTrue)},
    {CKA_ENCRYPT, &IsTrue, sizeof(IsTrue)},
    {CKA_MODULUS, modulus, sizeof(modulus)},
    {CKA_PUBLIC_EXPONENT, exponent, sizeof(exponent)}
};
```

[к содержанию ↑](#)

Объект CKO\_PUBLIC\_KEY, тип CKK\_ECDSA (определен стандартом)

Объекты открытого ключа ECDSA (объект CKO\_PUBLIC\_KEY, тип CKK\_GENERIC\_SECRET) содержат открытые ключи ECDSA.

Объект CKO\_PUBLIC\_KEY типа CKK\_ECDSA содержит как общие атрибуты объектов класса [CKO\\_PUBLIC\\_KEY](#), так и специфические, которые приведены в следующей таблице.

Атрибуты объекта CKO\_PUBLIC\_KEY, тип CKK\_ECDSA

| Атрибут                                                      | Тип данных | Значение                                                  |
|--------------------------------------------------------------|------------|-----------------------------------------------------------|
| Атрибуты открытого ключа ECDSA (ECDSA Public Key Attributes) |            |                                                           |
| CKA_ECDSA_PARAMS <sup>1,2,3</sup>                            | Byte array | Значение ECPParameters из стандарта X9.62 в DER-кодировке |
| CKA_EC_POINT <sup>1,3,4</sup>                                | Byte array | Значение ECPoint из стандарта X9.62 в DER-кодировке       |

- <sup>1</sup> должен быть определен при создании объекта с помощью функции **C\_CreateObject**.
- <sup>2</sup> должен быть определен при генерации объекта с помощью функций **C\_GenerateKey** или **C\_GenerateKeyPair**
- <sup>3</sup> должен остаться *незаданным* при расшифровании объекта с помощью функции **C\_UnwrapKey**

<sup>4</sup> должен остаться *незаданным* при генерации объекта с помощью функций **C\_GenerateKey** или **C\_GenerateKeyPair**

Больше информации по ключам ECDSA можно найти в PKCS#11.

Шаблон создания открытого ключа ECDSA

Шаблон создания открытого ECDSA ключа

```
CK_OBJECT_CLASS class = CKO_PUBLIC_KEY;
CK_KEY_TYPE keyType = CKK_ECDSA;
CK_CHAR label[] = "An ECDSA public key object";
CK_BYTE ecdsaParams[] = {... };
CK_BYTE ecPoint[] = {... };

CK_BBOOL true = TRUE;
CK_ATTRIBUTE template[] = {
    {CKA_CLASS, &class, sizeof(class)}
    ,
    {CKA_KEY_TYPE, &keyType, sizeof(keyType)}
    ,
    {CKA_TOKEN, &true, sizeof(true)}
    ,
    {CKA_LABEL, label, sizeof(label)}
    ,
    {CKA_ECDSA_PARAMS, ecdsaParams, sizeof(ecdsaParams)}
    ,
    {CKA_EC_POINT, ecPoint, sizeof(ecPoint)}
};
```

[к содержанию ↑](#)

Объект CKO\_PUBLIC\_KEY, тип CKK\_GOSTR3410 (определен стандартом)

Объекты открытого ключа CKO\_PUBLIC\_KEY типа CKK\_GOSTR3410 содержат открытые ключи ГОСТ Р 34.10-2001 или ГОСТ Р 34.10-2012 длиной 512 бит.

Объект CKO\_PUBLIC\_KEY типа CKK\_GOSTR3410 содержит как общие атрибуты объектов класса [CKO\\_PUBLIC\\_KEY](#), так и специфические, которые приведены в следующей таблице.

Атрибуты объекта CKO\_PUBLIC\_KEY, тип CKK\_GOSTR3410

| Атрибут                                                                                            | Тип данных | Значение                                                                                                                                                                                                                                                 |
|----------------------------------------------------------------------------------------------------|------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Атрибуты открытого ключа ГОСТ Р 34.10-2001 / ГОСТ Р 34.10-2012 (GOST R 3410 Public Key Attributes) |            |                                                                                                                                                                                                                                                          |
| CKA_VALUE1,2                                                                                       | Byte array | Открытый ключ длиной 64 байта: 32 байта для каждой координаты X и Y точки (X, Y) на эллиптической кривой в порядке, начиная с младшего байта (little endian)                                                                                             |
| CKA_GOSTR3410PARAMS1,3                                                                             | Byte array | Идентификатор, указывающий на тип объекта данных ГОСТ Р 34.10-2001 (OID параметра) в DER-кодировке. Когда ключ использует домен, параметр объекта ключа типа CKK_GOSTR3410 должен быть указан с тем же самым атрибутом CKA_OBJECT_ID                     |
| CKA_GOSTR3411PARAMS1,3,4                                                                           | Byte array | Идентификатор, указывающий на тип объекта данных ГОСТ Р 34.11-94 или ГОСТ Р 34.11-2012 (OID параметра) в DER-кодировке. Когда ключ использует домен, параметр объекта ключа типа CKK_GOSTR3411 должен быть указан с тем же самым атрибутом CKA_OBJECT_ID |

|                                   |            |                                                                                                                                                                                                                                                                           |
|-----------------------------------|------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| CKA_GOST28147_PARAMS <sup>4</sup> | Byte array | Идентификатор, указывающий на тип объекта данных ГОСТ 28147-89 (OID параметра) в DER-кодировке. Когда ключ использует домен, параметр объекта ключа типа CKK_GOSTR28147 должен быть указан с тем же самым атрибутом CKA_OBJECT_ID. Значение атрибута может быть пропущено |
|-----------------------------------|------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

<sup>1</sup> должен быть определен при создании объекта с помощью функции **C\_CreateObject**.

<sup>2</sup> должен остаться *незаданным* при генерации объекта с помощью функций **C\_GenerateKey** или **C\_GenerateKeyPair**

<sup>3</sup> должен быть определен при генерации объекта с помощью функций **C\_GenerateKey** или **C\_GenerateKeyPair**

<sup>4</sup> может быть модифицирован после создания объекта с помощью вызова функции **C\_SetAttributeValue**.

Устройства Рутокен, сертифицированные ФСБ, не поддерживают создание (импорт) ключей функцией C\_CreateObject по алгоритмам ГОСТ 28147-89, ГОСТ 34.10-2001 и ГОСТ 34.10-2012 в долговременную память (с флагом CKA\_TOKEN = TRUE).

#### Шаблон создания открытого ключа ГОСТ Р 34.10-2001

```
CK_OBJECT_CLASS your_class = CKO_PUBLIC_KEY;
CK_KEY_TYPE keyType = CKK_GOSTR3410;
CK_UTF8CHAR label[] = "A GOST R34.10-2001 public_key object";
CK_BYTE keyPairIdGost[] = {"GOST R 34.10-2001 sample key pair 1 ID (Aktiv Co.)"};
CK_BYTE gostR3410params_oid[] = {0x06, 0x07, 0x2a, 0x85, 0x03, 0x02, 0x02, 0x23, 0x00};
CK_BYTE gostR3411params_oid[] = {0x06, 0x07, 0x2a, 0x85, 0x03, 0x02, 0x02, 0x1e, 0x00};
CK_BYTE gost28147params_oid[] = {0x06, 0x07, 0x2a, 0x85, 0x03, 0x02, 0x02, 0x1f, 0x00};
CK_BYTE parametersGost28147[] = {0x06, 0x07, 0x2a, 0x85, 0x03, 0x02, 0x02, 0x1f, 0x00};
CK_BYTE value[64] = {...};
CK_BBOOL IsTrue = CK_TRUE;
CK_BBOOL IsFalse = CK_FALSE;

CK_ATTRIBUTE template[] = {
    {CKA_CLASS, &your_class, sizeof(your_class)},
    {CKA_KEY_TYPE, &keyType, sizeof(keyType)},
    {CKA_TOKEN, &IsTrue, sizeof(IsTrue)},
    {CKA_LABEL, label, sizeof(label)-1},
    {CKA_ID, &keyPairIdGost, sizeof(keyPairIdGost)-1},
    {CKA_PRIVATE, &IsFalse, sizeof(IsFalse)},
    {CKA_GOSTR3410PARAMS, gostR3410params_oid, sizeof(gostR3410params_oid)},
};
```

#### Шаблон создания открытого ключа ГОСТ Р 34.10-2012 (длина закрытого ключа 256 бит)

```
CK_OBJECT_CLASS your_class = CKO_PUBLIC_KEY;
CK_KEY_TYPE keyType = CKK_GOSTR3410;
CK_UTF8CHAR label[] = "A GOST R34.10-2012 public_key object";
CK_BYTE keyPairIdGost_256[] = {"GOST R 34.10-2012(256) sample key pair (Aktiv Co.)"};
CK_BYTE parametersGostR3410[] = {0x06, 0x07, 0x2a, 0x85, 0x03, 0x02, 0x02, 0x23, 0x01};
CK_BYTE parametersGostR3411_256[] = {0x06, 0x08, 0x2a, 0x85, 0x03, 0x07, 0x01, 0x01, 0x02, 0x02};
CK_BYTE parametersGost28147[] = {0x06, 0x07, 0x2a, 0x85, 0x03, 0x02, 0x02, 0x1f, 0x00};
CK_BYTE value[64] = {...};
CK_BBOOL IsTrue = CK_TRUE;
CK_BBOOL IsFalse = CK_FALSE;

CK_ATTRIBUTE template[] = {
    {CKA_CLASS, &your_class, sizeof(your_class)},
    {CKA_KEY_TYPE, &keyType, sizeof(keyType)},
    {CKA_TOKEN, &IsTrue, sizeof(IsTrue)},
    {CKA_LABEL, label, sizeof(label)-1},
    {CKA_ID, &keyPairIdGost_256, sizeof(keyPairIdGost_256)-1},
    {CKA_PRIVATE, &IsFalse, sizeof(IsFalse)},
    {CKA_GOSTR3410PARAMS, parametersGostR3410, sizeof(parametersGostR3410)},
    {CKA_GOSTR3411PARAMS, parametersGostR3411_256, sizeof(parametersGostR3411_256)},
    {CKA_VALUE, value, sizeof(value)}
};
```

## Объект SKO\_PUBLIC\_KEY, тип CKK\_GOSTR3410\_512 (определен расширением стандарта)

Объекты открытого ключа ГОСТ Р 34.10-2012 (объект SKO\_PUBLIC\_KEY, тип CKK\_GOSTR3410\_512) содержат открытые ключи ГОСТ Р 34.10-2012 длиной 1024 бита.

Объект SKO\_PUBLIC\_KEY типа CKK\_GOSTR3410 содержит как общие атрибуты объектов класса [SKO\\_PUBLIC\\_KEY](#), так и специфические, которые приведены в следующей таблице.

### Атрибуты объекта SKO\_PUBLIC\_KEY, тип CKK\_GOSTR3410

| Атрибут                                    | Тип данных | Значение                                                                                                                                                                                                                                                                  |
|--------------------------------------------|------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Атрибуты открытого ключа ГОСТ Р 34.10-2012 |            |                                                                                                                                                                                                                                                                           |
| SKA_VALUE <sup>1,2</sup>                   | Byte array | Открытый ключ длиной 128 байт: 64 байта для каждой координаты X и Y точки (X, Y) на эллиптической кривой в порядке, начиная с младшего байта (little endian)                                                                                                              |
| SKA_GOSTR3410PARAMS <sup>1,3</sup>         | Byte array | Идентификатор, указывающий на тип объекта данных ГОСТ Р 34.10-2012 (OID параметра) в DER-кодировке. Когда ключ использует домен, параметр объекта ключа типа CKK_GOSTR3410 должен быть указан с тем же самым атрибутом SKA_OBJECT_ID                                      |
| SKA_GOSTR3411PARAMS <sup>1,3,4</sup>       | Byte array | Идентификатор, указывающий на тип объекта данных ГОСТ Р 34.11-2012 (OID параметра) в DER-кодировке. Когда ключ использует домен, параметр объекта ключа типа CKK_GOSTR3411 должен быть указан с тем же самым атрибутом SKA_OBJECT_ID                                      |
| SKA_GOST28147_PARAMS <sup>4</sup>          | Byte array | Идентификатор, указывающий на тип объекта данных ГОСТ 28147-89 (OID параметра) в DER-кодировке. Когда ключ использует домен, параметр объекта ключа типа CKK_GOSTR28147 должен быть указан с тем же самым атрибутом SKA_OBJECT_ID. Значение атрибута может быть пропущено |

<sup>1</sup> должен быть определен при создании объекта с помощью функции **C\_CreateObject**.

<sup>2</sup> должен остаться *незаданным* при генерации объекта с помощью функций **C\_GenerateKey** или **C\_GenerateKeyPair**

<sup>3</sup> должен быть определен при генерации объекта с помощью функций **C\_GenerateKey** или **C\_GenerateKeyPair**

<sup>4</sup> может быть модифицирован после создания объекта с помощью вызова функции **C\_SetAttributeValue**.

Устройства Рутокен, сертифицированные ФСБ, не поддерживают создание (импорт) ключей функцией C\_CreateObject по алгоритмам ГОСТ 28147-89, ГОСТ 34.10-2001 и ГОСТ 34.10-2012 в долговременную память (с флагом SKA\_TOKEN = TRUE).

#### Шаблон создания открытого ключа ГОСТ Р 34.10-2012 (длина закрытого ключа 512 бит)

```
CK_OBJECT_CLASS your_class = CKO_PUBLIC_KEY;
CK_KEY_TYPE keyType = CKK_GOSTR3410_512;
CK_UTF8CHAR label[] = "A GOST R34.10-2012 public_key object";
CK_BYTE keyPairIdGost_512[] = {"GOST R 34.10-2012(512) sample key pair (Aktiv Co.)"};
CK_BYTE parametersGostR3410_512[] = {0x06, 0x09, 0x2a, 0x85, 0x03, 0x07, 0x01, 0x02, 0x01, 0x02, 0x01};
CK_BYTE parametersGostR3411_512[] = {0x06, 0x08, 0x2a, 0x85, 0x03, 0x07, 0x01, 0x01, 0x02, 0x03};
CK_BYTE parametersGost28147[] = {0x06, 0x07, 0x2a, 0x85, 0x03, 0x02, 0x02, 0x1f, 0x00};
CK_BYTE value[128] = {...};
CK_BBOOL IsTrue = CK_TRUE;
CK_BBOOL IsFalse = CK_FALSE;

CK_ATTRIBUTE template[] = {
    {CKA_CLASS, &your_class, sizeof(your_class)},
    {CKA_KEY_TYPE, &keyType, sizeof(keyType)},
    {CKA_TOKEN, &IsTrue, sizeof(IsTrue)},
    {CKA_LABEL, label, sizeof(label)-1},
    {CKA_ID, &keyPairIdGost_512, sizeof(keyPairIdGost_512)-1},
    {CKA_PRIVATE, &IsFalse, sizeof(IsFalse)},
    {CKA_GOSTR3410PARAMS, parametersGostR3410_512, sizeof(parametersGostR3410_512)},
    {CKA_GOSTR3411PARAMS, parametersGostR3411_512, sizeof(parametersGostR3411_512)},
    {CKA_VALUE, value, sizeof(value)}
};
```

[к содержанию ↑](#)

## Объекты закрытых ключей

- Объект **CKO\_PRIVATE\_KEY**, тип **CKK\_RSA** (определен стандартом)
- Объект **CKO\_PRIVATE\_KEY**, тип **CKK\_ECDSA** (определен стандартом)
- Объект **CKO\_PRIVATE\_KEY**, тип **CKK\_GOSTR3410** (определен стандартом)
- Объект **CKO\_PRIVATE\_KEY**, тип **CKK\_GOSTR3410\_512** (определен расширением стандарта)

Объекты закрытых ключей (класс объектов **CKO\_PRIVATE\_KEY**) содержат закрытые ключи. Следующая таблица определяет общие атрибуты для всех закрытых ключей, в дополнение к **общим атрибутам**, определенным для этого класса объектов.

#### Общие атрибуты закрытых ключей

| Атрибут                                                               | Тип данных | Значение                                                                                                                                         |
|-----------------------------------------------------------------------|------------|--------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Общие атрибуты закрытых ключей (Common Private Key Attributes)</b> |            |                                                                                                                                                  |
| CKA_SUBJECT <sup>1</sup>                                              | Byte array | Имя владельца сертификата в DER-кодировке (по умолчанию пусто)                                                                                   |
| CKA_SENSITIVE <sup>1,2</sup>                                          | CK_BBOOL   | CK_TRUE, если ключ является чувствительным (не может быть извлечен из токена в открытом виде) <sup>3</sup>                                       |
| CKA_DECRYPT <sup>1</sup>                                              | CK_BBOOL   | CK_TRUE, если ключ поддерживает расшифрование <sup>3</sup>                                                                                       |
| CKA_SIGN <sup>1</sup>                                                 | CK_BBOOL   | CK_TRUE, если ключ поддерживает формирование подписи, которая представлена в виде приложения к данным <sup>3</sup>                               |
| CKA_SIGN_RECOVER <sup>1</sup>                                         | CK_BBOOL   | CK_TRUE, если ключ поддерживает формирование подписи с восстановлением данных <sup>3</sup>                                                       |
| CKA_UNWRAP <sup>1</sup>                                               | CK_BBOOL   | CK_TRUE, если ключ поддерживает расшифрование ключей (размаскирование, т.е. может быть использован для расшифрования других ключей) <sup>3</sup> |
| CKA_EXTRACTABLE <sup>1,4</sup>                                        | CK_BBOOL   | CK_TRUE, если ключ является извлекаемым и может быть зашифрован <sup>3</sup>                                                                     |

|                                                                                                               |                  |                                                                                                                                                                                                                                            |
|---------------------------------------------------------------------------------------------------------------|------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| CKA_ALWAYS_SENSITIVE <sup>5,6,7</sup>                                                                         | CK_BBOOL         | CK_TRUE, если ключ <i>всегда</i> имеет значение атрибута CKA_SENSITIVE равным CK_TRUE                                                                                                                                                      |
| CKA_NEVER_EXTRACTABLE <sup>5,6,7</sup>                                                                        | CK_BBOOL         | CK_TRUE, если ключ <i>никогда</i> не имеет значение атрибута CKA_EXTRACTABLE равным CK_TRUE                                                                                                                                                |
| CKA_WRAP_WITH_TRUSTED <sup>2</sup>                                                                            | CK_BBOOL         | CK_TRUE, если ключ может быть зашифрован только с помощью ключа шифрования ключей со значением атрибута CKA_TRUSTED равным CK_TRUE. По умолчанию имеет значение CK_FALSE.                                                                  |
| CKA_UNWRAP_TEMPLATE                                                                                           | CK_ATTRIBUTE_PTR | Для ключей шифрования ключей. Шаблон атрибутов для всех ключей, которые могут быть расшифрованы с помощью ключа шифрования ключей. Число атрибутов в массиве равно отношению переменной атрибута <i>ulValueLen</i> к размеру CK_ATTRIBUTE. |
| CKA_ALWAYS_AUTHENTICATE                                                                                       | CK_BBOOL         | Если CK_TRUE, то для каждого действия с ключом требуется ввод PIN-кода пользователя. По умолчанию имеет значение CK_FALSE.                                                                                                                 |
| <b>Атрибуты закрытых ключей, определенные производителем (Rutoken Vendors Defined Private Key Attributes)</b> |                  |                                                                                                                                                                                                                                            |
| CKA_CAPI_ID                                                                                                   |                  | Идентификатор ключевой пары (по умолчанию 0).                                                                                                                                                                                              |
| CKA_PUBLIC_KEY_RSF_ID                                                                                         |                  | Идентификатор RSF-файла, хранящего соответствующий открытый ключ (по умолчанию 0).                                                                                                                                                         |
| CKA_PRIVATE_KEY_RSF_ID                                                                                        |                  | Идентификатор RSF-файла, хранящего закрытый ключ (по умолчанию 0).                                                                                                                                                                         |
| CKA_VENDOR_KEY_PIN_ENTER                                                                                      | CK_BBOOL         | Если CK_TRUE, то при каждой операции подписи или шифрования/расшифрования данным ключом требуется ввод PIN-кода пользователя на экране РутOKEN PINPad.                                                                                     |
| CKA_VENDOR_KEY_CONFIRM_OP                                                                                     | CK_BBOOL         | Если CK_TRUE, то при каждой операции или шифрования/расшифрования требуется подтверждение на экране РутOKEN PINPad.                                                                                                                        |
| CKA_VENDOR_KEY_JOURNAL                                                                                        | CK_BBOOL         | Если CK_TRUE, то ключ может использоваться только для работы с журналом РутOKEN PINPad                                                                                                                                                     |

<sup>1</sup> может быть изменен после создания объекта с помощью функции **C\_SetAttributeValue**

<sup>2</sup> после установки атрибута в значение CK\_TRUE он становится read-only

<sup>3</sup> значение по умолчанию определяется РутOKEN и может зависеть от значения других атрибутов

<sup>4</sup> после установки атрибута в значение CK\_FALSE он становится read-only

<sup>5</sup> должен остаться *незаданным* при создании объекта с помощью функции **C\_CreateObject**

<sup>6</sup> должен остаться *незаданным* при создании объекта с помощью функций **C\_GenerateKey** или **C\_GenerateKeyPair**

<sup>7</sup> должен остаться *незаданным* при расшифровании ключа с помощью функции **C\_UnwrapKey**

В целях совместимости значения полей CKA\_SUBJECT и CKA\_ID закрытого ключа должны совпадать с полями CKA\_SUBJECT и CKA\_ID соответствующего сертификата и открытого ключа. Однако стандарт не настаивает на выполнении этого требования, так же как и не является обязательным хранение сертификата и закрытого ключа на токене.

Если атрибут **CKA\_SENSITIVE** имеет значение CK\_TRUE или атрибут **CKA\_EXTRACTABLE** имеет значение CK\_FALSE, то некоторые атрибуты закрытого ключа не могут быть извлечены из памяти токена в виде открытого текста. Такие атрибуты определяются отдельно для каждого типа закрытого ключа.

Атрибут **CKA\_ALWAYS\_AUTHENTICATE** может быть использован для аутентификации пользователя (ввода PIN-кода пользователя) при каждом использовании закрытого ключа. «Использование» в этом случае означает выполнение какой-либо криптографической операции, например, создание подписи или дешифрование. Этот атрибут может иметь значение CK\_TRUE, только когда **CKA\_PRIVATE** тоже имеет значение CK\_TRUE.

Повторная аутентификация выполняется вызовом функции **C\_Login** со значением *userType*, равным **CKU\_CONTEXT\_SPECIFIC**, непосредственно после инициализации криптографической операции, использующей ключ (например, функцией **C\_SignInit**). В этом случае тип пользователя задается неявно, в зависимости от параметров ключа. Если функция **C\_Login** возвращает CKR\_OK, то аутентификация пользователя прошла успешно, и ключ находится в аутентифицированном состоянии до тех пор, пока криптографическая операция не завершится, успешно или безуспешно (например, с помощью функций **C\_Sign**, **C\_SignFinal**). Возвращаемый функцией **C\_Login** код ошибки CKR\_PIN\_INCORRECT означает, что пользователю отказано в доступе к ключу и результат выполнения криптографической операции будет таким же, как если бы функция **C\_Login** не была вызвана. В обоих случаях состояние сессии будет оставаться одинаковым, однако повторные неудачные попытки выполнить аутентификацию могут привести к блокировке PIN-кода. В таком случае **C\_Login** вернет код ошибки CKR\_PIN\_LOCKED и выполнит выход пользователя с устройства Рутокен. Ошибочная попытка аутентификации или отсутствие аутентификации при атрибуте CKA\_ALWAYS\_AUTHENTICATE равным CK\_TRUE вызовет ошибку CKR\_USER\_NOT\_LOGGED\_IN при использовании ключа. Функция **C\_Login** вернет ошибку CKR\_OPERATION\_NOT\_INITIALIZED, которая не повлияет на выполнение текущей криптографической операции, если повторная аутентификация будет совершена при значении атрибута CKA\_ALWAYS\_AUTHENTICATE равного CK\_FALSE.

Атрибут **CKA\_CAPI\_ID** предназначен для хранения ID контейнера из библиотеки **rtCSP**. В случае, когда нет возможности выяснить значения атрибутов **CKA\_CAPI\_ID**, **CKA\_PUBLIC\_KEY\_RSF\_ID** и **CKA\_PRIVATE\_KEY\_RSF\_ID**, следует использовать значение по умолчанию, равное 0.

Атрибуты **CKA\_VENDOR\_KEY\_PIN\_ENTER**, **CKA\_VENDOR\_KEY\_CONFIRM\_OP** и **CKA\_VENDOR\_KEY\_JOURNAL** применимы только для Рутокен PINPad.

[к содержанию ↑](#)

## Объект CKO\_PRIVATE\_KEY, тип CKK\_RSA (определен стандартом)

Объекты закрытого ключа RSA (объект CKO\_PRIVATE\_KEY, тип CKK\_GENERIC\_SECRET) содержат закрытые ключи RSA.

Объект CKO\_PRIVATE\_KEY типа CKK\_RSA содержит как общие атрибуты объектов класса [CKO\\_PRIVATE\\_KEY](#), так и специфические, которые приведены в следующей таблице.

### Атрибуты объекта CKO\_PRIVATE\_KEY, тип CKK\_RSA

| Атрибут                                                   | Тип данных  | Значение                                                                             |
|-----------------------------------------------------------|-------------|--------------------------------------------------------------------------------------|
| Атрибуты закрытого ключа RSA (RSA Private Key Attributes) |             |                                                                                      |
| CKA_MODULUS <sup>1,2,3</sup>                              | Big integer | Модуль <i>n</i>                                                                      |
| CKA_PUBLIC_EXPONENT <sup>2,3</sup>                        | Big integer | Открытая экспонента <i>e</i>                                                         |
| CKA_PRIVATE_EXPONENT <sup>1,2,3,4</sup>                   | Big integer | Закрытая экспонента <i>d</i>                                                         |
| CKA_PRIME_1 <sup>2,3,4</sup>                              | Big integer | Простое число <i>p</i>                                                               |
| CKA_PRIME_2 <sup>2,3,4</sup>                              | Big integer | Простое число <i>q</i>                                                               |
| CKA_EXPONENT_1 <sup>2,3,4</sup>                           | Big integer | Закрытая экспонента <i>d</i> по модулю <i>p</i> -1                                   |
| CKA_EXPONENT_2 <sup>2,3,4</sup>                           | Big integer | Закрытая экспонента <i>d</i> по модулю <i>q</i> -1                                   |
| CKA_COEFFICIENT <sup>2,3,4</sup>                          | Big integer | Коэффициент CTR (КТО, Китайская теорема об остатках) RSA $q^{-1}$ по модулю <i>p</i> |

<sup>1</sup> должен быть определен при создании объекта с помощью функции **C\_CreateObject**.

<sup>2</sup> должен остаться *незаданным* при генерации объекта с помощью функций **C\_GenerateKey** или **C\_GenerateKeyPair**

<sup>3</sup> должен остаться *незаданным* при расшифровании объекта с помощью функции **C\_UnwrapKey**

<sup>4</sup> не может быть *раскрыт* если объект имеет атрибут **CKA\_SENSITIVE**, установленным в положение CK\_TRUE или атрибут **CKA\_EXTRACTABLE** в положение CK\_FALSE.

Больше информации по ключам RSA можно найти в PKCS #1.

## Шаблон создания закрытого ключа RSA

```
CK_OBJECT_CLASS your_class = CKO_PRIVATE_KEY;
CK_KEY_TYPE keyType = CKK_RSA;
CK_UTF8CHAR label[] = "An RSA private_key object";
CK_BYTE subject[] = {...};
CK_BYTE id[] = {123};
CK_BYTE modulus[] = {...};
CK_BYTE publicExponent[] = {...};
CK_BYTE privateExponent[] = {...};
CK_BYTE prime1[] = {...};
CK_BYTE prime2[] = {...};
CK_BYTE exponent1[] = {...};
CK_BYTE exponent2[] = {...};
CK_BYTE coefficient[] = {...};
CK_BBOOL IsTrue = CK_TRUE;

CK_ATTRIBUTE template[] = {
    {CKA_CLASS, &your_class, sizeof(your_class)},
    {CKA_KEY_TYPE, &keyType, sizeof(keyType)},
    {CKA_TOKEN, &IsTrue, sizeof(IsTrue)},
    {CKA_LABEL, label, sizeof(label)-1},
    {CKA_SUBJECT, subject, sizeof(subject)},
    {CKA_ID, id, sizeof(id)},
    {CKA_SENSITIVE, &IsTrue, sizeof(IsTrue)},
    {CKA_DECRYPT, &IsTrue, sizeof(IsTrue)},
    {CKA_SIGN, &IsTrue, sizeof(IsTrue)},
    {CKA_MODULUS, modulus, sizeof(modulus)},
    {CKA_PUBLIC_EXPONENT, publicExponent, sizeof(publicExponent)},
    {CKA_PRIVATE_EXPONENT, privateExponent, sizeof(privateExponent)},
    {CKA_PRIME_1, prime1, sizeof(prime1)},
    {CKA_PRIME_2, prime2, sizeof(prime2)},
    {CKA_EXPONENT_1, exponent1, sizeof(exponent1)},
    {CKA_EXPONENT_2, exponent2, sizeof(exponent2)},
    {CKA_COEFFICIENT, coefficient, sizeof(coefficient)}
};
```

[к содержанию ↑](#)

## Объект CKO\_PRIVATE\_KEY, тип CKK\_ECDSA (определен стандартом)

Объекты закрытого ключа ECDSA (объект CKO\_PRIVATE\_KEY, тип CKK\_GENERIC\_SECRET) содержат закрытые ключи ECDSA.

Объект CKO\_PRIVATE\_KEY типа CKK\_ECDSA содержит как общие атрибуты объектов класса [CKO\\_PRIVATE\\_KEY](#), так и специфические, которые приведены в следующей таблице.

### Атрибуты объекта CKO\_PRIVATE\_KEY, тип CKK\_ECDSA

| Атрибут                                                              | Тип данных  | Значение                                  |
|----------------------------------------------------------------------|-------------|-------------------------------------------|
| <b>Атрибуты закрытого ключа ECDSA (ECDSA Private Key Attributes)</b> |             |                                           |
| CKA_ECDSA_PARAMS <sup>1,3,4</sup>                                    | Byte array  | Значение X9.62 параметров в DER-кодировке |
| CKA_VALUE <sup>1,3,4,5</sup>                                         | Big integer | Значение закрытого ключа                  |

<sup>1</sup> должен быть определен при создании объекта с помощью функции **C\_CreateObject**.

<sup>2</sup> должен быть определен при генерации объекта с помощью функций **C\_GenerateKey** или **C\_GenerateKeyPair**

<sup>3</sup> должен остаться *незаданным* при расшифровании объекта с помощью функции **C\_UnwrapKey**

<sup>4</sup> должен остаться *незаданным* при генерации объекта с помощью функций **C\_GenerateKey** или **C\_GenerateKeyPair**

<sup>5</sup> не может быть раскрыт если объект имеет атрибут **CKA\_SENSITIVE**, установленным в положение CK\_TRUE или атрибут **CKA\_EXTRACTABLE** в положение CK\_FALSE.

Больше информации по ключам ECDSA можно найти в PKCS#11.

Шаблон создания закрытого ключа ECDSA

```
CK_OBJECT_CLASS class = CKO_PRIVATE_KEY;
CK_KEY_TYPE keyType = CKK_ECDSA;
CK_UTF8CHAR label[] = "An ECDSA private key object";
CK_BYTE subject[] = {... };
CK_BYTE id[] = { 123 };
CK_BYTE ecdsaParams[] = {... };
CK_BYTE value[] = {... };

CK_BBOOL true = TRUE;
CK_ATTRIBUTE template[] = {
    {CKA_CLASS, &class, sizeof(class)}
    ,
    {CKA_KEY_TYPE, &keyType, sizeof(keyType)}
    ,
    {CKA_TOKEN, &true, sizeof(true)}
    ,
    {CKA_LABEL, label, sizeof(label) - 1}
    ,
    {CKA_SUBJECT, subject, sizeof(subject)}
    ,
    {CKA_ID, id, sizeof(id)}
    ,
    {CKA_SENSITIVE, &true, sizeof(true)}
    ,
    {CKA_DERIVE, &true, sizeof(true)}
    ,
    {CKA_ECDSA_PARAMS, ecdsaParams, sizeof(ecdsaParams)}
    ,
    {CKA_VALUE, value, sizeof(value)}
};
```

[к содержанию ↑](#)

## Объект CKO\_PRIVATE\_KEY, тип CKK\_GOSTR3410 (определен стандартом)

Объекты закрытого ключа типа ГОСТ Р 34.10 (объект CKO\_PRIVATE\_KEY, тип CKK\_GOSTR3410) содержат закрытые ключи ГОСТ Р 34.10-2001 или ГОСТ Р 34.10-2012 длиной 256 бит.

Объект CKO\_PRIVATE\_KEY типа CKK\_GOSTR3410 содержит как общие атрибуты объектов класса [CKO\\_PRIVATE\\_KEY](#), так и специфические, которые приведены в следующей таблице.

### Атрибуты объекта CKO\_PRIVATE\_KEY, тип CKK\_GOSTR3410

| Атрибут                                                                                             | Тип данных | Значение                                                                                                                                                                                                                             |
|-----------------------------------------------------------------------------------------------------|------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Атрибуты закрытого ключа ГОСТ Р 34.10-2001 / ГОСТ Р 34.10-2012 (GOST R 3410 Private Key Attributes) |            |                                                                                                                                                                                                                                      |
| CKA_VALUE1,2,3,4                                                                                    | Byte array | Закрытый ключ длиной 32 байта в порядке, начиная с младшего байта (little endian)                                                                                                                                                    |
| CKA_GOSTR3410PARAMS1,2,3                                                                            | Byte array | Идентификатор, указывающий на тип объекта данных ГОСТ Р 34.10-2001 (OID параметра) в DER-кодировке. Когда ключ использует домен, параметр объекта ключа типа CKK_GOSTR3410 должен быть указан с тем же самым атрибутом CKA_OBJECT_ID |

|                                        |            |                                                                                                                                                                                                                                                                           |
|----------------------------------------|------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| CKA_GOSTR3411PARAMS <sup>1,2,3,5</sup> | Byte array | Идентификатор, указывающий на тип объекта данных ГОСТ Р 34.11-94 или ГОСТ Р 34.11-2012 (OID параметра) в DER-кодировке. Когда ключ использует домен, параметр объекта ключа типа CKK_GOSTR3411 должен быть указан с тем же самым атрибутом CKA_OBJECT_ID                  |
| CKA_GOST28147_PARAMS <sup>2,3,5</sup>  | Byte array | Идентификатор, указывающий на тип объекта данных ГОСТ 28147-89 (OID параметра) в DER-кодировке. Когда ключ использует домен, параметр объекта ключа типа CKK_GOSTR28147 должен быть указан с тем же самым атрибутом CKA_OBJECT_ID. Значение атрибута может быть пропущено |

<sup>1</sup> должен быть определен при создании объекта с помощью функции **C\_CreateObject**.

<sup>2</sup> должен остаться *незаданным* при генерации объекта с помощью функций **C\_GenerateKey** или **C\_GenerateKeyPair**

<sup>3</sup> должен остаться *незаданным* при расшифровании объекта с помощью функции **C\_UnwrapKey**

<sup>4</sup> не может быть раскрыт если объект имеет атрибут **CKA\_SENSITIVE**, установленным в положение CK\_TRUE или атрибут **CKA\_EXTRACTABLE** в положение CK\_FALSE.

<sup>5</sup> может быть модифицирован после создания объекта с помощью вызова функции **C\_SetAttributeValue**.

Обратите внимание, что при генерации закрытого ключа типа ГОСТ Р 34.10-2001/2012 доменные параметры не указываются в шаблоне ключа. Это обусловливается тем, что закрытые ключи этого типа генерируются только как часть ключевой пары ГОСТ Р 34.10-2001/2012 и доменные параметры для пары указываются в шаблоне открытого ключа.

Устройства Рутокен, сертифицированные ФСБ, не поддерживают создание (импорт) ключей функцией C\_CreateObject по алгоритмам ГОСТ 28147-89, ГОСТ 34.10-2001 и ГОСТ 34.10-2012 в долговременную память (с флагом CKA\_TOKEN = TRUE).

#### Шаблон создания закрытого ключа ГОСТ Р 34.10-2001

```
CK_OBJECT_CLASS your_class = CKO_PRIVATE_KEY;
CK_KEY_TYPE keyType = CKK_GOSTR3410;
CK_UTF8CHAR label[] = "A GOST R34.10-2001 private_key object";
CK_BYTE keyPairIdGost[] = {"GOST R 34.10-2001 sample key pair 1 ID (Aktiv Co.)"};
CK_BYTE gostR3410params_oid[] = {0x06, 0x07, 0x2a, 0x85, 0x03, 0x02, 0x02, 0x23, 0x00};
CK_BYTE gostR3411params_oid[] = {0x06, 0x07, 0x2a, 0x85, 0x03, 0x02, 0x02, 0x1e, 0x00};
CK_BYTE gost28147params_oid[] = {0x06, 0x07, 0x2a, 0x85, 0x03, 0x02, 0x02, 0x1f, 0x00};
CK_BYTE value[32] = {...};
CK_BBOOL IsTrue = CK_TRUE;

CK_ATTRIBUTE template[] = {
    {CKA_CLASS, &your_class, sizeof(your_class)},
    {CKA_KEY_TYPE, &keyType, sizeof(keyType)},
    {CKA_TOKEN, &IsTrue, sizeof(IsTrue)},
    {CKA_LABEL, label, sizeof(label)-1},
    {CKA_ID, &keyPairIdGost, sizeof(keyPairIdGost)-1},
    {CKA_PRIVATE, &IsTrue, sizeof(IsTrue)},
    {CKA_GOSTR3410PARAMS, gostR3410params_oid, sizeof(gostR3410params_oid)},
    {CKA_GOSTR3411PARAMS, gostR3411params_oid, sizeof(gostR3411params_oid)},
    {CKA_VALUE, value, sizeof(value)}
};
```

#### Шаблон создания открытого ключа ГОСТ Р 34.10-2012 (256 бит)

```
CK_OBJECT_CLASS your_class = CKO_PRIVATE_KEY;
CK_KEY_TYPE keyType = CKK_GOSTR3410;
CK_UTF8CHAR label[] = "A GOST R34.10-2012 private_key object";
CK_BYTE keyPairIdGost_256[] = {"GOST R 34.10-2012(256) sample key pair (Aktiv Co.)"};
CK_BYTE parametersGostR3410[] = {0x06, 0x07, 0x2a, 0x85, 0x03, 0x02, 0x02, 0x23, 0x01};
CK_BYTE parametersGostR3411_256[] = {0x06, 0x08, 0x2a, 0x85, 0x03, 0x07, 0x01, 0x01, 0x02, 0x02};
CK_BYTE parametersGost28147[] = {0x06, 0x07, 0x2a, 0x85, 0x03, 0x02, 0x02, 0x1f, 0x00};
CK_BYTE value[32] = {...};
CK_BBOOL IsTrue = CK_TRUE;
CK_ATTRIBUTE template[] = {
    {CKA_CLASS, &your_class, sizeof(your_class)},
    {CKA_KEY_TYPE, &keyType, sizeof(keyType)},
    {CKA_TOKEN, &IsTrue, sizeof(IsTrue)},
    {CKA_LABEL, label, sizeof(label)-1},
    {CKA_ID, &keyPairIdGost_256, sizeof(keyPairIdGost_256)-1},
    {CKA_PRIVATE, &IsTrue, sizeof(IsTrue)},
    {CKA_GOSTR3410PARAMS, parametersGostR3410, sizeof(parametersGostR3410)},
    {CKA_GOSTR3411PARAMS, parametersGostR3411_256, sizeof(parametersGostR3411_256)},
    {CKA_VALUE, value, sizeof(value)}
};
```

[к содержанию ↑](#)

## Объект CKO\_PRIVATE\_KEY, тип CKK\_GOSTR3410\_512 (определен расширением стандарта)

Объекты закрытого ключа CKO\_PRIVATE\_KEY типа CKK\_GOSTR3410 содержат закрытые ключи ГОСТ Р 34.10-2012 длиной 512 бит.

Объект CKO\_PRIVATE\_KEY типа CKK\_GOSTR3410 содержит как общие атрибуты объектов класса [CKO\\_PRIVATE\\_KEY](#), так и специфические, которые приведены в следующей таблице.

#### Атрибуты объекта CKO\_PRIVATE\_KEY, тип CKK\_GOSTR3410

| Атрибут                                                                                                    | Тип данных | Значение                                                                                                                                                                                                                                                                  |
|------------------------------------------------------------------------------------------------------------|------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Атрибуты закрытого ключа ГОСТ Р 34.10-2001 / ГОСТ Р 34.10-2012 (GOST R 3410 Private Key Attributes)</b> |            |                                                                                                                                                                                                                                                                           |
| CKA_VALUE <sup>1,2,3,4</sup>                                                                               | Byte array | Закрытый ключ длиной 64 байта в порядке, начиная с младшего байта (little endian)                                                                                                                                                                                         |
| CKA_GOSTR3410PARAMS <sup>1,2,3</sup>                                                                       | Byte array | Идентификатор, указывающий на тип объекта данных ГОСТ Р 34.10-2012 (OID параметра) в DER-кодировке. Когда ключ использует домен, параметр объекта ключа типа CKK_GOSTR3410 должен быть указан с тем же самым атрибутом CKA_OBJECT_ID                                      |
| CKA_GOSTR3411PARAMS <sup>1,2,3,5</sup>                                                                     | Byte array | Идентификатор, указывающий на тип объекта данных ГОСТ Р 34.11-2012 (OID параметра) в DER-кодировке. Когда ключ использует домен, параметр объекта ключа типа CKK_GOSTR3411 должен быть указан с тем же самым атрибутом CKA_OBJECT_ID                                      |
| CKA_GOST28147_PARAMS <sup>2,3,5</sup>                                                                      | Byte array | Идентификатор, указывающий на тип объекта данных ГОСТ 28147-89 (OID параметра) в DER-кодировке. Когда ключ использует домен, параметр объекта ключа типа CKK_GOSTR28147 должен быть указан с тем же самым атрибутом CKA_OBJECT_ID. Значение атрибута может быть пропущено |

<sup>1</sup> должен быть определен при создании объекта с помощью функции **C\_CreateObject**.

<sup>2</sup> должен остаться *незаданным* при генерации объекта с помощью функций **C\_GenerateKey** или **C\_GenerateKeyPair**

- <sup>3</sup> должен остаться *незаданным* при расшифровании объекта с помощью функции **C\_UnwrapKey**
- <sup>4</sup> не может быть раскрыт если объект имеет атрибут **CKA\_SENSITIVE**, установленным в положение CK\_TRUE или атрибут **CKA\_EXTRACTABLE** в положение CK\_FALSE.
- <sup>5</sup> может быть модифицирован после создания объекта с помощью вызова функции **C\_SetAttributeValue**.

Обратите внимание, что при генерации закрытого ключа типа ГОСТ Р 34.10-2012 доменные параметры не указываются в шаблоне ключа. Это обусловливается тем, что закрытые ключи этого типа генерируются только как часть ключевой пары и доменные параметры для пары указываются в шаблоне открытого ключа ГОСТ Р 34.10-2012.

Шаблон создания закрытого ключа ГОСТ Р 34.10-2012 (512 бит)

```
CK_OBJECT_CLASS your_class = CKO_PRIVATE_KEY;
CK_KEY_TYPE keyType = CKK_GOSTR3410_512;
CK_UTF8CHAR label[] = {"A GOST R34.10-2012 private_key object"};
CK_BYTE keyPairIdGost_512[] = {"GOST R 34.10-2012(512) sample key pair (Aktiv Co.)"};
CK_BYTE parametersGostR3410_512[] = {0x06, 0x09, 0x2a, 0x85, 0x03, 0x07, 0x01, 0x02, 0x01, 0x02, 0x02, 0x01};
CK_BYTE parametersGostR3411_512[] = {0x06, 0x08, 0x2a, 0x85, 0x03, 0x07, 0x01, 0x01, 0x02, 0x03 };
CK_BYTE parametersGost28147[] = {0x06, 0x07, 0x2a, 0x85, 0x03, 0x02, 0x02, 0x1f, 0x00};
CK_BYTE value[64] = {...};
CK_BBOOL IsTrue = CK_TRUE;

CK_ATTRIBUTE template[] = {
    {CKA_CLASS, &your_class, sizeof(your_class)},
    {CKA_KEY_TYPE, &keyType, sizeof(keyType)},
    {CKA_TOKEN, &IsTrue, sizeof(IsTrue)},
    {CKA_LABEL, label, sizeof(label)-1},
    {CKA_ID, &keyPairIdGost_512, sizeof(keyPairIdGost_512)-1},
    {CKA_PRIVATE, &IsTrue, sizeof(IsTrue)},
    {CKA_GOSTR3410PARAMS, parametersGostR3410_512, sizeof(parametersGostR3410_512)},
    {CKA_GOSTR3411PARAMS, parametersGostR3411_512, sizeof(parametersGostR3411_512)},
    {CKA_VALUE, value, sizeof(value)}
};
```

[к содержанию ↑](#)

## Объекты секретных ключей

- Объект **CKO\_SECRET\_KEY**, тип **CKK\_GENERIC\_SECRET** (определен стандартом)
- Объект **CKO\_SECRET\_KEY**, тип **CKK\_GOST** (определен производителем)
- Объект **CKO\_SECRET\_KEY**, тип **CKK\_GOST28147** (определен стандартом)

Объекты секретных ключей (класс объектов **CKO\_SECRET\_KEY**) содержат секретные ключи. Следующая таблица определяет общие атрибуты для всех секретных ключей в дополнение к **общим атрибутам**, определенным для этого класса объектов.

### Общие атрибуты секретных ключей

| Атрибут                                                        | Тип данных | Значение                                                                                                                                |
|----------------------------------------------------------------|------------|-----------------------------------------------------------------------------------------------------------------------------------------|
| Общие атрибуты секретных ключей (Common Secret Key Attributes) |            |                                                                                                                                         |
| CKA_SENSITIVE <sup>1,2</sup>                                   | CK_BBOOL   | CK_TRUE, если объект является чувствительным (не может быть извлечен из токена в открытом виде). По умолчанию имеет значение CK_FALSE   |
| CKA_ENCRYPT <sup>1</sup>                                       | CK_BBOOL   | CK_TRUE, если ключ поддерживает шифрование <sup>3</sup>                                                                                 |
| CKA_DECRYPT <sup>1</sup>                                       | CK_BBOOL   | CK_TRUE, если ключ поддерживает расшифрование <sup>3</sup>                                                                              |
| CKA_SIGN <sup>1</sup>                                          | CK_BBOOL   | CK_TRUE, если ключ поддерживает создание подписи (код аутентификации), где подпись представлена в виде приложения к данным <sup>3</sup> |

|                                                                                                               |                  |                                                                                                                                                                                                                                             |
|---------------------------------------------------------------------------------------------------------------|------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| CKA_VERIFY <sup>1</sup>                                                                                       | CK_BBOOL         | CK_TRUE, если ключ поддерживает проверку подписи (код аутентификации), где подпись представлена в виде приложения к данным <sup>3</sup>                                                                                                     |
| CKA_WRAP <sup>1</sup>                                                                                         | CK_BBOOL         | CK_TRUE, если поддерживает шифрование ключей (маскирование, т.е. может быть использован для шифрования других ключей) <sup>3</sup>                                                                                                          |
| CKA_UNWRAP <sup>1</sup>                                                                                       | CK_BBOOL         | CK_TRUE, если ключ поддерживает расшифрование ключей (размаскирование, т.е. может быть использован для расшифрования других ключей) <sup>3</sup>                                                                                            |
| CKA_EXTRACTABLE <sup>1,4</sup>                                                                                | CK_BBOOL         | CK_TRUE если ключ является извлекаемым и может быть зашифрован <sup>3</sup>                                                                                                                                                                 |
| CKA_ALWAYS_SENSITIVE <sup>5,6,7</sup>                                                                         | CK_BBOOL         | CK_TRUE, если ключ <i>всегда</i> имеет значение атрибута CKA_SENSITIVE равным CK_TRUE                                                                                                                                                       |
| CKA_NEVER_EXTRACTABLE <sup>5,6,7</sup>                                                                        | CK_BBOOL         | CK_TRUE, если ключ <i>никогда</i> не имеет значение атрибута CKA_EXTRACTABLE равным CK_TRUE                                                                                                                                                 |
| CKA_CHECK_VALUE                                                                                               | Byte array       | Контрольная сумма ключа                                                                                                                                                                                                                     |
| CKA_WRAP_WITH_TRUSTED <sup>2</sup>                                                                            | CK_BBOOL         | CK_TRUE, если ключ может быть зашифрован только с помощью ключа шифрования со значением атрибута CKA_TRUSTED равным CK_TRUE. По умолчанию имеет значение CK_FALSE.                                                                          |
| CKA_TRUSTED <sup>8</sup>                                                                                      | CK_BBOOL         | Ключ шифрования может быть использован для шифрования ключей со значением атрибута CKA_WRAP_WITH_TRUSTED равным CK_TRUE.                                                                                                                    |
| CKA_WRAP_TEMPLATE                                                                                             | CK_ATTRIBUTE_PTR | Для ключей шифрования ключей. Шаблон атрибутов для всех ключей, которые зашифрованы с помощью данного ключа шифрования. Число атрибутов в массиве равно отношению переменной <i>ulValueLen</i> атрибута к размеру CK_ATTRIBUTE.             |
| CKA_UNWRAP_TEMPLATE                                                                                           | CK_ATTRIBUTE_PTR | Для ключей шифрования ключей. Шаблон атрибутов для всех ключей, которые могут быть расшифрованы с помощью данного ключа шифрования. Число атрибутов в массиве равно отношению переменной <i>ulValueLen</i> атрибута к размеру CK_ATTRIBUTE. |
| <b>Атрибуты секретных ключей, определенные производителем (Rutoken Vendors Defined Secret Key Attributes)</b> |                  |                                                                                                                                                                                                                                             |
| CKA_SECRET_KEY_RSF_ID                                                                                         |                  | Идентификатор RSF-файла, хранящего секретный ключ (по умолчанию 0)                                                                                                                                                                          |

<sup>1</sup> может быть изменен после создания объекта с помощью функции **C\_SetAttributeValue**

<sup>2</sup> после установки атрибута в значение CK\_TRUE он становится read-only

<sup>3</sup> значение по умолчанию определяется Рутoken и может зависеть от значения других атрибутов

<sup>4</sup> после установки атрибута в значение CK\_FALSE он становится read-only

<sup>5</sup> должен остаться *незаданным* при создании объекта с помощью функции **C\_CreateObject**

<sup>6</sup> должен остаться *незаданным* при создании объекта с помощью функций **C\_GenerateKey** или **C\_GenerateKeyPair**

<sup>7</sup> должен остаться *незаданным* при расшифровании ключа с помощью функции **C\_UnwrapKey**

<sup>8</sup> значение CK\_TRUE может быть задано только Администратором

Если атрибут **CKA\_SENSITIVE** имеет значение CK\_TRUE или атрибут **CKA\_EXTRACTABLE** имеет значение CK\_FALSE, то некоторые атрибуты секретного ключа не могут быть извлечены из памяти токена в виде открытого текста. Такие атрибуты определяются отдельно для каждого типа секретного ключа.

Для объектов симметричных ключей атрибут значения проверки ключа (key check value — KCV) называется **CKA\_CHECK\_VALUE**, имеет данные типа массив байтов размером в 3 байта и работает подобно контрольной сумме ключа. Эти данные используются для перекрестной проверки симметричных ключей в системах, где применяются аналогичные ключи, а также для проверки корректности ключа при вводе его вручную или восстановлении из резервной копии.

Свойства:

1. Для двух криптографически идентичных ключей значения этого атрибута должны быть одинаковы.
2. Атрибут CKA\_CHECK\_VALUE не должен быть годен для восстановления какой-либо части значения ключа.
3. Неуникальность. Два разных ключа могут иметь одинаковые значения атрибута CKA\_CHECK\_VALUE. Это маловероятно, но возможно.

Атрибут CKA\_CHECK\_VALUE является необязательным, но если поддерживается, то его значение всегда поставляется библиотекой независимо от того, как объект ключа был создан или получен. Атрибут может поставляться даже в случае запрета на выполнение операции шифрования для ключа (то есть когда CKA\_ENCRYPT имеет значение CK\_FALSE).

Если значение поставляется в шаблоне приложения (допускается, но не является обязательным), то оно должно соответствовать значению, вычисляемому библиотекой; в противном случае библиотека вернет ошибку CKR\_ATTRIBUTE\_VALUE\_INVALID.

Генерация значения проверки ключа может быть предотвращена, если атрибут в шаблоне задан как «не имеющий значения» (нулевой длины). Приложение может запросить значение атрибута в любое время с помощью функции C\_GetAttributeValue. Функция C\_SetAttributeValue может быть использована для уничтожения атрибута, задав его как «не имеющий значения».

Если иное не указано в определении объекта, значение этого атрибута выводится из объекта ключа путем взятия первых трех байтов блока, состоящего из нулевых байтов (0x00) и зашифрованного способом и режимом по умолчанию (ECB, electronic codebook), ассоциированными с типом объекта секретного ключа.

В случае, когда нет возможности выяснить значение атрибута CKA\_SECRET\_KEY\_RSF\_ID, следует использовать значение по умолчанию, равное 0.

Атрибуты CKA\_VENDOR\_KEY\_PIN\_ENTER и CKA\_VENDOR\_KEY\_CONFIRM\_OP применимы только для Рутокен PINPad.

[к содержанию ↑](#)

## Объект CKO\_SECRET\_KEY, тип CKK\_GENERIC\_SECRET (определен стандартом)

Объекты общего секретного ключа (объект CKO\_SECRET\_KEY, тип CKK\_GENERIC\_SECRET) содержат абстрактные симметричные ключи, по сути своей представляющие данные произвольной длины.

Ключи этого типа не поддерживают операции шифрования и расшифрования, однако из них могут быть получены другие ключи и они могут использоваться в операциях HMAC. Таким образом, механизм использования этих ключей пользователь определяет сам.

Объект CKO\_SECRET\_KEY типа CKK\_GENERIC\_SECRET содержит как общие атрибуты объектов класса CKO\_SECRET\_KEY, так и специфические.

### Атрибуты объекта CKO\_SECRET\_KEY, тип CKK\_GENERIC\_SECRET

| Атрибут                                                             | Тип данных | Значение                              |
|---------------------------------------------------------------------|------------|---------------------------------------|
| Атрибуты общего симметричного ключа (Generic Secret Key Attributes) |            |                                       |
| CKA_VALUE <sup>1,2,3,4</sup>                                        | Byte array | Значение (тело) ключа случайной длины |
| CKA_VALUE_LEN <sup>5,6</sup>                                        | CK_ULONG   | Длина значения ключа в байтах         |

<sup>1</sup> должен быть определен при создании объекта с помощью функции C\_CreateObject.

<sup>2</sup> должен остаться *незаданным* при генерации объекта с помощью функций C\_GenerateKey или C\_GenerateKeyPair

<sup>3</sup> должен остаться *незаданным* при расшифровании объекта с помощью функции C\_UnwrapKey

<sup>4</sup> не может быть раскрыт если объект имеет атрибут CKA\_SENSITIVE, установленным в положение CK\_TRUE или атрибут CKA\_EXTRACTABLE в положение CK\_FALSE.

<sup>5</sup> должен остаться *незаданным* при создании объекта с помощью функции C\_CreateObject

<sup>6</sup> должен быть определен при генерации объекта с помощью функций C\_GenerateKey или C\_GenerateKeyPair

Шаблон создания общего симметричного ключа

```
CK_OBJECT_CLASS your_class = CKO_SECRET_KEY;
CK_KEY_TYPE keyType = CKK_GENERIC_SECRET;
CK_UTF8CHAR label[] = "A generic secret key object";
CK_BYTE value[] = {...};
CK_BBOOL IsTrue = CK_TRUE;

CK_ATTRIBUTE template[] = {
    {CKA_CLASS, &your_class, sizeof(your_class)},
    {CKA_KEY_TYPE, &keyType, sizeof(keyType)},
    {CKA_TOKEN, &IsTrue, sizeof(IsTrue)},
    {CKA_LABEL, label, sizeof(label)-1},
    {CKA_DERIVE,, &IsTrue, sizeof(IsTrue)},
    {CKA_VALUE, value, sizeof(value)}
};
```

[к содержанию ↑](#)

Объект CKO\_SECRET\_KEY, тип CKK\_GOST (определен производителем)

Симметричный ключ CKO\_SECRET\_KEY типа CKK\_GOST содержит общие атрибуты для объектов класса [CKO\\_SECRET\\_KEY](#). Специфические (определенные пользователем) атрибуты для типа CKK\_GOST приведены в таблице выше

Объект CKO\_SECRET\_KEY типа CKK\_GOST поддерживается библиотеками [rtPKCS11](#) до версии 2.30.

Атрибуты объекта CKO\_SECRET\_KEY, тип CKK\_GOST

| Атрибут                                                                                    | Тип данных | Значение                                                                                                   |
|--------------------------------------------------------------------------------------------|------------|------------------------------------------------------------------------------------------------------------|
| Атрибуты, определенные производителем (Rutoken Vendors Defined CKK_GOST Object Attributes) |            |                                                                                                            |
| CKA_VALUE                                                                                  | Byte Array | Тело ключа                                                                                                 |
| CKA_GOST_KEY_OPTIONS                                                                       |            | Опции ключа ГОСТ 28147-89 (режим шифрования: простая замена, гаммирование, гаммирование с обратной связью) |
| CKA_GOST_KEY_FLAGS                                                                         |            | Флаги ключа ГОСТ 28147-89                                                                                  |

[к содержанию ↑](#)

Объект CKO\_SECRET\_KEY, тип CKK\_GOST28147 (определен стандартом)

Симметричный ключ GOST 28147-89 (объект CKO\_SECRET\_KEY, тип CKK\_GOST28147) содержит как общие атрибуты для объектов класса [CKO\\_SECRET\\_KEY](#), так и специфические, которые приведены в таблице ниже.

Объект CKO\_SECRET\_KEY типа CKK\_GOST28147 поддерживается библиотекой [rtPKCS11](#) начиная с версии 2.30.

Атрибуты объекта CKO\_SECRET\_KEY, тип CKK\_GOST28147

| Атрибут                                                                                    | Тип данных | Значение                                                                                                                                                                                                       |
|--------------------------------------------------------------------------------------------|------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Атрибуты ГОСТ 28147 (GOST 28147 Attributes)                                                |            |                                                                                                                                                                                                                |
| CKA_VALUE <sup>1,2,3,4</sup>                                                               | Byte array | Тело ключа размером в 32 байта, записанное в порядке, начиная со младшего байта                                                                                                                                |
| CKA_GOST28147_PARAMS <sup>1,5,6</sup>                                                      | Byte array | Идентификатор типа объекта данных ГОСТ 28147 (OID параметра) в DER-кодировке. Когда ключ использует домен, параметр объекта ключа типа CKK_GOST28147 должен быть указан с тем же самым атрибутом CKA_OBJECT_ID |
| Атрибуты, определенные производителем (Rutoken Vendors Defined CKK_GOST Object Attributes) |            |                                                                                                                                                                                                                |

|                        |                                                   |
|------------------------|---------------------------------------------------|
| CKA_SECRET_KEY_RSFS_ID | Идентификатор RSF-файла, хранящего секретный ключ |
|------------------------|---------------------------------------------------|

<sup>1</sup> должен быть определен при создании объекта с помощью функции **C\_CreateObject**.

<sup>2</sup> должен остаться *незаданным* при генерации объекта с помощью функций **C\_GenerateKey** или **C\_GenerateKeyPair**

<sup>3</sup> должен остаться *незаданным* при расшифровании объекта с помощью функции **C\_UnwrapKey**

<sup>4</sup> не может быть раскрыт если объект имеет атрибут CKA\_SENSITIVE, установленным в положение CK\_TRUE или атрибут CKA\_EXTRACTABLE в положение CK\_FALSE.

<sup>5</sup> должен быть определен при генерации объекта с помощью функций **C\_GenerateKey** или **C\_GenerateKeyPair**

<sup>6</sup> должен быть определен при расшифровании объекта с помощью функции **C\_UnwrapKey**

Устройства Рутокен, сертифицированные ФСБ, не поддерживают создание (импорт) ключей функцией C\_CreateObject по алгоритмам ГОСТ 28147-89, ГОСТ 34.10-2001 и ГОСТ 34.10-2012 в долговременную память (с флагом CKA\_TOKEN = TRUE).

#### Шаблон создания симметричного ключа ГОСТ 28147-89

```
CK_OBJECT_CLASS your_class = CKO_SECRET_KEY;
CK_KEY_TYPE keyType = CKK_GOST28147;
CK_UTF8CHAR label[] = "A GOST 28147-89 secret key object";
CK_BYTE value[32] = {...};
CK_BYTE parametersGost28147[] = {0x06, 0x07, 0x2a, 0x85, 0x03, 0x02, 0x02, 0x1f, 0x01};
CK_BBOOL IsTrue = CK_TRUE;

CK_ATTRIBUTE template[] = {
    {CKA_CLASS, &your_class, sizeof(your_class)},
    {CKA_KEY_TYPE, &keyType, sizeof(keyType)},
    {CKA_TOKEN, &IsTrue, sizeof(IsTrue)},
    {CKA_PRIVATE, &IsTrue, sizeof(IsTrue)},
    {CKA_LABEL, label, sizeof(label)-1},
    {CKA_ENCRYPT, &IsTrue, sizeof(IsTrue)},
    {CKA_DECRYPT, &IsTrue, sizeof(IsTrue)},
    {CKA_GOST28147PARAMS, parametersGost28147, sizeof(parametersGost28147)},
    {CKA_VALUE, value, sizeof(value)}
};
```

[к содержанию ↑](#)