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 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[1,2]	CK_KEY_TYPE	Тип ключа
CKA_ID[3]	Byte array	Идентификатор ключа (по умолчанию пусто)
CKA_START_DATE[3]	CK_DATE	Дата начала действия ключа (по умолчанию пусто)
CKA_END_DATE[3]	CK_DATE	Дата окончания действия ключа (по умолчанию пусто)
CKA_DERIVE[3]	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[4,5,6]	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[4,5,6]	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.

[1] должен быть определен при создании объекта с помощью функции **C_CreateObject.**

[2] должен быть определен при расшифровании объекта с помощью функции **C_UnwrapKey.**

[3] может быть изменен после создания объекта вызовом функции **C_SetAttributeValue** или при копировании объекта вызовом функции **C_CopyObject**

[4] должен остаться *незаданным* при создании объекта с помощью функции **C_CreateObject**

[5] должен остаться *незаданным* при генерации объекта с помощью функций **C_GenerateKey** или **C_GenerateKeyPair**

[6] должен остаться *незаданным* при расшифровании объекта с помощью функции **C_UnwrapKey**

Поле **CKA_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 **CKA_ID** определяется приложением.

Атрибуты **CKA_START_DATE** и **CKA_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 **CKA_DERIVE** имеет значение CK_TRUE в том и только в том случае, если из этого ключа могут быть получены другие ключи.

Атрибут **CKA_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 **CKA_KEY_GEN_MECHANISM** идентифицирует механизм генерации ключа с использованием данных ключа. Он содержит действующее значение, только если атрибут **CKA_LOCAL** имеет значение CK_TRUE. Если **CKA_LOCAL** имеет значение CK_FALSE, значение атрибута будет CK_UNAVAILABLE_INFORMATION.

к содержанию ↑

Объекты открытых ключей

Объекты открытых ключей (класс объектов **CKO_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)		
CKA_SUBJECT[1]	Byte array	Имя ключа в DER-кодировке (по умолчанию пусто)
CKA_ENCRYPT[1]	CK_BBOOL	CK_TRUE, если ключ поддерживает шифрование[2]
CKA_VERIFY[1]	CK_BBOOL	CK_TRUE, если ключ поддерживает проверку подписи, представленной в виде приложения к данным[2]
CKA_VERIFY_RECOVER[1]	CK_BBOOL	CK_TRUE, если ключ поддерживает проверку подписи с восстановлением (восстанавливаемую из данных)[2]
CKA_WRAP[1]	CK_BBOOL	CK_TRUE, если ключ поддерживает маскирование других ключей (то есть может быть использован для шифрования других ключей)[2]

CKA_TRUSTED[3]	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 *ulValueLen* атрибута к размеру CK_ATTRIBUTE.
Атрибуты открытых ключей, определенные производителем (Rutoken Vendors Defined Public Key Attributes)		
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 использован только для работы с журналом Рутокен PINPad

[1] может быть изменен после создания объекта с помощью функции **C_SetAttributeValue**

[2] значение по умолчанию определяется токеном и может зависеть от значения других атрибутов

[3] 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 Рутокен 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[1,2]	Big integer	Модуль *n*
CKA_MODULUS_BITS[3,4]	CK_ULONG	Длина модуля *n* в битах
CKA_PUBLIC_EXPONENT[1]	Big integer	Открытая экспонента *e*

[1] должен быть определен при создании объекта с помощью функции **C_CreateObject.**

[2] должен остаться *незаданным* при генерации объекта с помощью функций **C_GenerateKey** или **C_GenerateKeyPair**

[3] должен остаться *незаданным* при создании объекта с помощью функции **C_CreateObject**

[4] 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[1,2,3]	Byte array	Значение ECParameters из стандарта X9.62 в DER-кодировке
CKA_EC_POINT[1,3,4]	Byte array	Значение ECPoint из стандарта X9.62 в DER-кодировке

[1] должен быть определен при создании объекта с помощью функции **C_CreateObject.**

[2] должен быть определен при генерации объекта с помощью функций **C_GenerateKey** или **C_GenerateKeyPair**

[3] должен остаться *незаданным* при расшифровании объекта с помощью функции **C_UnwrapKey**

[4] 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_VALUE[1,2]	Byte array	Открытый ключ длиной 64 байта: 32 байта для каждой координаты X и У точки (X, У) на эллиптической кривой в порядке, начиная с младшего байта (little endian)
CKA_GOSTR3410PARAMS[1,3]	Byte array	Идентификатор, указывающий на тип объекта данных ГОСТ Р 34.10-2001 (OID парамсета) 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[1,3,4]	Byte array	Идентификатор, указывающий на тип объекта данных ГОСТ Р 34.11-94 или ГОСТ Р 34.11-2012 (OID парамсета) 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_GOST 28147_PARAMS[4]	Byte array	Идентификатор, указывающий на тип объекта данных ГОСТ 28147-89 (OID парамсета) 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

[1] должен быть определен при создании объекта с помощью функции **C_CreateObject.**

[2] должен остаться *незаданным* при генерации объекта с помощью функций **C_GenerateKey** или **C_GenerateKeyPair**

[3] должен быть определен при генерации объекта с помощью функций **C_GenerateKey** или **C_GenerateKeyPair**

[4] 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)}
};
```

к содержанию ↑

Объект CKO_PUBLIC_KEY, тип CKK_GOSTR3410_512 (определен расширением стандарта)

Объекты открытого ключа ГОСТ Р 34.10-2012 (объект CKO_PUBLIC_KEY, тип CKK_GOSTR3410_512) содержат открытые ключи ГОСТ Р 34.10-2012 длиной 1024 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 34.10-2012		
CKA_VALUE[1,2]	Byte array	Открытый ключ длиной 128 байт: 64 байта для каждой координаты X и У точки (X, У) на эллиптической кривой в порядке, начиная с младшего байта (little endian)
CKA_GOSTR3410PARAMS[1,3]	Byte array	Идентификатор, указывающий на тип объекта данных ГОСТ Р 34.10-2012 (OID парамсета) 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[1,3,4]	Byte array	Идентификатор, указывающий на тип объекта данных ГОСТ Р 34.11-2012 (OID парамсета) 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[4]	Byte array	Идентификатор, указывающий на тип объекта данных ГОСТ 28147-89 (OID парамсета) 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

[1] должен быть определен при создании объекта с помощью функции **C_CreateObject.**

[2] должен остаться *незаданным* при генерации объекта с помощью функций **C_GenerateKey** или **C_GenerateKeyPair**

[3] должен быть определен при генерации объекта с помощью функций **C_GenerateKey** или **C_GenerateKeyPair**

[4] 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 34.10-2012 (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(keyPairIdGost1_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
Общие атрибуты закрытых ключей (Common Private Key Attributes)		
CKA_SUBJECT [1]	Byte array	Имя владельца сертификата в DER-кодировке (по умолчанию пусто)
CKA_SENSITIVE [1,2]	CK_BBOOL	CK_TRUE, если ключ является чувствительным (не может быть извлечен из токена в открытом виде)[3]
CKA_DECRYPT [1]	CK_BBOOL	CK_TRUE, если ключ поддерживает расшифрование[3]
CKA_SIGN [1]	CK_BBOOL	CK_TRUE, если ключ поддерживает формирование подписи, которая представлена в виде приложения к данным[3]
CKA_SIGN_RECOVER [1]	CK_BBOOL	CK_TRUE, если ключ поддерживает формирование подписи с восстановлением данных[3]
CKA_UNWRAP [1]	CK_BBOOL	CK_TRUE, если ключ поддерживает расшифрование ключей (размаскирование, т.е. может быть использован для расшифрования других ключей)[3]
CKA_EXTRACTABLE [1,4]	CK_BBOOL	CK_TRUE,если ключ является извлекаемым и может быть зашифрован[3]

CKA_ALWAYS_SENSITIVE[5,6,7]	CK_BBOOL	CK_TRUE, если ключ *всегда* имеет значение атрибута CKA_SENSITIVE равным CK_TRUE
CKA_NEVER_EXTRACTABLE[5,6,7]	CK_BBOOL	CK_TRUE, если ключ *никогда* не имеет значение атрибута CKA_EXTRACTABLE равным CK_TRUE
CKA_WRAP_WITH_TRUSTED[2]	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 ключа. Число атрибутов в массиве равно отношению переменной атрибута *ulValueLen* 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.
Атрибуты закрытых ключей, определенные производителем (Rutoken Vendors Defined Private Key Attributes)		
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 Рутокен PINPad.
CKA_VENDOR_KEY_CONFIRM_OP	CK_BBOOL	Если CK_TRUE, то при каждой операции или шифрования/расшифрования требуется подтверждение на экране Рутокен PINPad.
CKA_VENDOR_KEY_JOURNAL	CK_BBOOL	Если CK_TRUE, то ключ может использован только для работы с журналом Рутокен PINPad

[1] может быть изменен после создания объекта с помощью функции **C_SetAttributeValue**

[2] после установки атрибута в значение CK_TRUE он становится read-only

[3] значение по умолчанию определяется Рутокен и может зависеть от значения других атрибутов

[4] после установки атрибута в значение CK_FALSE он становится read-only

[5] должен остаться *незаданным* при создании объекта с помощью функции **C_CreateObject**

[6] должен остаться *незаданным* при создании объекта с помощью функций **C_GenerateKey** или **C_GenerateKeyPair**

[7] 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 токенав 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 аутентификации 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[1,2,3]	Big integer	Модуль *n*
CKA_PUBLIC_EXPONENT[2,3]	Big integer	Открытая экспонента *e*
CKA_PRIVATE_EXPONENT[1,2,3,4]	Big integer	Закрытая экспонента *d*
CKA_PRIME_1[2,3,4]	Big integer	Простое число *p*
CKA_PRIME_2[2,3,4]	Big integer	Простое число *q*
CKA_EXPONENT_1[2,3,4]	Big integer	Закрытая экспонента *d* по модулю *p*-1
CKA_EXPONENT_2[2,3,4]	Big integer	Закрытая экспонента *d* по модулю *q*-1
CKA_COEFFICIENT[2,3,4]	Big integer	Коэффициент CTR (КТО, Китайская теорема об остатках) RSA q^{-1} по модулю *p*

[1] должен быть определен при создании объекта с помощью функции **C_CreateObject.**

[2] должен остаться *незаданным* при генерации объекта с помощью функций **C_GenerateKey** или **C_GenerateKeyPair**

[3] должен остаться *незаданным* при расшифровании объекта с помощью функции **C_UnwrapKey**

[4] 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
Атрибуты закрытого ключа ECDSA (ECDSA Private Key Attributes)		
CKA_ECDSA_PARAMS[1,3,4]	Byte array	Значение X9.62 параметров в DER-кодировке
CKA_VALUE[1,3,4,5]	Big integer	Значение закрытого ключа

[1]должен быть определен при создании объекта с помощью функции**C_CreateObject.**

[2] должен быть определен при генерации объекта с помощью функций**C_GenerateKey**или**C_GenerateKeyPair**

[3]должен остаться *незаданным* при расшифровании объекта с помощью функции **C_UnwrapKey**

[4] должен остаться *незаданным* при генерации объекта с помощью функций **C_GenerateKey** или **C_GenerateKeyPair**

[5] 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_VALUE [1,2,3,4]	Byte array	Закрытый ключ длиной 32 байта в порядке, начиная с младшего байта (little endian)
CKA_GOSTR3410PARAMS [1,2,3]	Byte array	Идентификатор, указывающий на тип объекта данных ГОСТ Р 34.10-2001 (OID парамсета) 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[1,2,3,5]	Byte array	Идентификатор, указывающий на тип объекта данных ГОСТ Р 34.11-94 или ГОСТ Р 34.11-2012 (OID парамсета) 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[2,3,5]	Byte array	Идентификатор, указывающий на тип объекта данных ГОСТ 28147-89 (OID парамсета) 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

[1] должен быть определен при создании объекта с помощью функции **C_CreateObject.**

[2] должен остаться *незаданным* при генерации объекта с помощью функций **C_GenerateKey** или **C_GenerateKeyPair**

[3] должен остаться *незаданным* при расшифровании объекта с помощью функции **C_UnwrapKey**

[4] не может быть раскрыт если объект имеет атрибут **CKA_SENSITIVE**, установленным в положение CK_TRUE или атрибут **CKA_EXTRACTABLE** в положение CK_FALSE.

[5] 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
Атрибуты закрытого ключа ГОСТ Р 34.10-2001 / ГОСТ Р 34.10-2012 (GOST R 3410 Private Key Attributes)		
CKA_VALUE[1,2,3,4]	Byte array	Закрытый ключ длиной 64 байта в порядке, начиная с младшего байта (little endian)
CKA_GOSTR3410PARAMS[1,2,3]	Byte array	Идентификатор, указывающий на тип объекта данных ГОСТ Р 34.10-2012 (OID парамсета) 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[1,2,3,5]	Byte array	Идентификатор, указывающий на тип объекта данных ГОСТ Р 34.11-2012 (OID парамсета) 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[2,3,5]	Byte array	Идентификатор, указывающий на тип объекта данных ГОСТ 28147-89 (OID парамсета) 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

[1] должен быть определен при создании объекта с помощью функции **C_CreateObject.**

[2] должен остаться *незаданным* при генерации объекта с помощью функций **C_GenerateKey** или **C_GenerateKeyPair**

[3] должен остаться *незаданным* при расшифровании объекта с помощью функции **C_UnwrapKey**

[4] не может быть раскрыт если объект имеет атрибут **CKA_SENSITIVE**, установленным в положение CK_TRUE или атрибут **CKA_EXTRACTABLE** в положение CK_FALSE.

[5] 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, 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

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[1,2]	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[1]	CK_BBOOL	CK_TRUE, если ключ поддерживает шифрование[3]
CKA_DECRYPT[1]	CK_BBOOL	CK_TRUE, если ключ поддерживает расшифрование[3]
CKA_SIGN[1]	CK_BBOOL	CK_TRUE, если ключ поддерживает создание подписи (код аутентификации), где подпись представлена в виде приложения к данным [3]

CKA_VERIFY[1]	CK_BBOOL	CK_TRUE, если ключ поддерживает проверку подписи (код аутентификации), где подпись представлена в виде приложения к данным [3]
CKA_WRAP[1]	CK_BBOOL	CK_TRUE, если поддерживает шифрование ключей (маскирование, т.е. может быть использован для шифрования других ключей)[3]
CKA_UNWRAP[1]	CK_BBOOL	CK_TRUE, если ключ поддерживает расшифрование ключей (размаскирование, т.е. может быть использован для расшифрования других ключей)[3]
CKA_EXTRACTABLE[1,4]	CK_BBOOL	CK_TRUE если ключ является извлекаемым и может быть зашифрован[3]
CKA_ALWAYS_SENSITIVE[5,6,7]	CK_BBOOL	CK_TRUE, если ключ *всегда* имеет значение атрибута CKA_SENSITIVE равным CK_TRUE
CKA_NEVER_EXTRACTABLE[5,6,7]	CK_BBOOL	CK_TRUE , если ключ *никогда* не имеет значение атрибута CKA_EXTRACTABLE равным CK_TRUE
CKA_CHECK_VALUE	Byte array	Контрольная сумма ключа
CKA_WRAP_WITH_TRUSTED[2]	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[8]	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 *ulValueLen* 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 *ulValueLen* атрибута к размеру CK_ATTRIBUTE.
Атрибуты секретных ключей, определенные производителем (Rutoken Vendors Defined Secret Key Attributes)		
CKA_SECRET_KEY_RSF_ID		Идентификатор RSF–файла, хранящего секретный ключ (по умолчанию 0)

[1] может быть изменен после создания объекта с помощью функции **C_SetAttributeValue**

[2] после установки атрибута в значение CK_TRUE он становится read-only

[3] значение по умолчанию определяется Рутокен и может зависеть от значения других атрибутов

[4] после установки атрибута в значение CK_FALSE он становится read-only

[5] должен остаться *незаданным* при создании объекта с помощью функции **C_CreateObject**

[6] должен остаться *незаданным* при создании объекта с помощью функций **C_GenerateKey** или **C_GenerateKeyPair**

[7] должен остаться *незаданным* при расшифровании ключа с помощью функции **C_UnwrapKey**

[8] 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[1,2,3,4]	Byte array	Значение (тело) ключа случайной длины
CKA_VALUE_LEN[5,6]	CK_ULONG	Длина значения ключа в байтах

[1] должен быть определен при создании объекта с помощью функции **C_CreateObject.**

[2] должен остаться *незаданным* при генерации объекта с помощью функций **C_GenerateKey** или **C_GenerateKeyPair**

[3] должен остаться *незаданным* при расшифровании объекта с помощью функции **C_UnwrapKey**

[4] не может быть раскрыт если объект имеет атрибут **CKA_SENSITIVE**, установленным в положение CK_TRUE или атрибут **CKA_EXTRACTABLE** в положение CK_FALSE.

[5] должен остаться *незаданным* при создании объекта с помощью функции **C_CreateObject**

[6] 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[1,2,3,4]	Byte array	Тело ключа размером в 32 байта, записанное в порядке, начиная со младшего байта
CKA_GOST28147_PARAMS[1,5,6]	Byte array	Идентификатор типа объекта данных ГОСТ 28147 (OID парамсета) 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_RSF_ID		Идентификатор RSF-файла, хранящего секретный ключ

[1] должен быть определен при создании объекта с помощью функции **C_CreateObject.**

[2] должен остаться *незаданным* при генерации объекта с помощью функций **C_GenerateKey** или **C_GenerateKeyPair**

[3] должен остаться *незаданным* при расшифровании объекта с помощью функции **C_UnwrapKey**

[4] не может быть раскрыт если объект имеет атрибут CKA_SENSITIVE,установленным в положение CK_TRUE или атрибут CKA_EXTRACTABLE в положение CK_FALSE.

[5] должен быть определен при генерации объекта с помощью функций **C_GenerateKey** или **C_GenerateKeyPair**

[6] 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 ↑