Объекты сертификатов (CKO_CERTIFICATE)

Этот раздел описывает класс объектов CKO_CERTIFICATE типа данных CK_OBJECT_CLASS, использующийся в атрибуте CKA_CLASS объектов.

Общие атрибуты объектов сертификатов

Объекты сертификатов (класс объектов **CKO_CERTIFICATE**) содержат сертификаты открытых ключей или сертификаты атрибутов. Следующая таблица определяет общие атрибуты для всех объектов сертификатов в дополнение к общим атрибутам, определенным для этого класса объектов.

Таблица 2.6. Общие атрибуты объектов сертификатов

Атрибут	Тип данных	Значение
Общие атрибуты объектов сертификатов (Common Certificate Object Attributes)		
CKA_CERTIFICATE_TYPE[1]	CK_CERTIFICATE_TYPE	Тип сертификата
CKA_TRUSTED[2]	CK_BBOOL	Сертификат является доверенным для приложения, которым был создан.
CKA_CERTIFICATE_CATEGORY	CK_ULONG	Категория сертификата: 0 = не определена (по умолчанию), 1 = пользователь токена, 2 = центр сертификации, 3 = другая категория
CKA_CHECK_VALUE	Byte array	Контрольная сумма
CKA_START_DATE	CK_DATE	Дата начала действия сертификата (по умолчанию пусто)
CKA_END_DATE	CK_DATE	Дата окончания действия сертификата (по умолчанию пусто)
Атрибуты объектов сертификатов, определенные производителем (Rutoken Vendors Defined Certificate Object Attributes)		
CKA_CAPI_ID		Идентификатор ключевой пары (по умолчанию 0)

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

[2] значение CK_TRUE может быть задано только Администратором

Атрибут **CKA_CERTIFICATE_TYPE** не может быть изменен после создания объекта. Устройства Рутокен поддерживают следующие типы сертификатов:

- сертификат открытого ключа X.509
- сертификат атрибута X.509

Значение CK_TRUE атрибута **CKA_TRUSTED** не может быть задано приложением и должно задаваться при инициализации токена или Администратором токена. Доверенный сертификат не может быть изменен.

Атрибут **CKA_CERTIFICATE_CATEGORY** используется для индикации типа хранимого сертификата: является ли он пользовательским с соответствующим ему закрытым ключом, хранимым на токене («пользователь токена»), сертификатом, выданным Центру Сертификации («центр сертификации») или сертификатом другой категории («другая категория»). Этот атрибут не может быть изменен после создания объекта.

Атрибуты **CKA_CERTIFICATE_CATEGORY** и **CKA_TRUSTED** используются совместно для отображения категорий сертификатов. Сертификат в certificates CDF (Certificate directory File, см. PKCS #15) будет помечен как сертификат категории «пользователь токена». Сертификат в trustedCertificates CDF или usefulCertificates CDF будет помечен как сертификат категории «центр сертификации» или «другие лица» в зависимости от значений атрибутов CommonCertificateAttribute.authority и CKA_TRUSTED, последний из которых показывает, к trustedCertificates или usefulCertificates CDF принадлежит сертификат.

Значение атрибута **CKA_CHECK_VALUE** выводится из самого сертификата следующим образом: берутся первые 3 байта SHA-1 хеша атрибута CKA_VALUE объекта сертификата.

Атрибуты **CKA_START_DATE** и **CKA_END_DATE** предназначены только для справочной информации; им не присваивается иного значения. Если они присутствуют, то приложение несет ответственность за установку значений, соответствующих полям сертификата «Действителен с» («not before») и «Действителен по» («not after») (если таковые имеются).

Атрибут **CKA_CAPI_ID** предназначен для хранения ID контейнера из библиотеки rtCSP. В случае, когда нет возможности выяснить значение атрибута, следует использовать значение по умолчанию, равное 0.

к содержанию ↑

Объекты сертификата открытого ключа X.509

Объекты сертификата X.509 (тип сертификата **CKC_X_509**) содержат сертификаты открытых ключей формата X.509. Следующая таблица определяет атрибуты объектов сертификата X.509 в дополнение к общим атрибутам, определенным для этого класса объектов.

Таблица 2.7. Атрибуты объектов сертификата X.509

Атрибут	Тип данных	Значение
CKA_SUBJECT[1]	Byte array	Имя владельца сертификата в DER-кодировке
CKA_ID	Byte array	Идентификатор ключа для пары «открытый/закрытый ключ» (по умолчанию пусто)
CKA_ISSUER	Byte array	Имя издателя сертификата в DER-кодировке (по умолчанию пусто)
CKA_SERIAL_NUMBER	Byte array	Серийный номер сертификата в DER-кодировке (по умолчанию пусто)
CKA_VALUE[2]	Byte array	Содержимое сертификата в BER-кодировке
CKA_URL[3]	RFC2279 string	Непустой атрибут задает URL, по которому можно получить сертификат целиком (по умолчанию пусто)
CKA_HASH_OF_SUBJECT_PUBLIC_KEY[4]	Byte array	Хеш SHA-1 открытого ключа владельца сертификата (по умолчанию пусто)
CKA_HASH_OF_ISSUER_PUBLIC_KEY[4]	Byte array	Хеш SHA-1 открытого ключа издателя сертификата (по умолчанию пусто)
CKA_JAVA_MIDP_SECURITY_DOMAIN	CK_ULONG	Домен безопасности Java MIDP: 0 = не определен (по умолчанию), 1 = производитель, 2 = оператор, 3 = третье лицо
CKA_NAME_HASH_ALGORITHM	CK_MECHANISM_TYPE	Определяет механизм, используемый для вычисления значений CKA_HASH_OF_SUBJECT_PUBLIC_KEY и CKA_HASH_OF_ISSUER_PUBLIC_KEY. Если атрибут не задан, то значение по умолчанию - SHA-1.

[1] должен быть задан при создании объекта.

[2] должен быть задан при создании объекта. Не может быть пустым, если CKA_URL пусто.

[3] не может быть пустым, если CKA_VALUE пусто.

[4] может быть пустым, только если CKA_URL пусто

Только атрибуты **CKA_ID**, **CKA_ISSUER** и **CKA_SERIAL_NUMBER** могут быть изменены после создания объекта.

Атрибут **CKA_ID** предназначен для различения нескольких пар «открытый/закрытый ключ», принадлежащих одному субъекту (вне зависимости от того, хранятся они на токене или нет). (Поскольку ключи отличаются также именами субъектов, возможно, что ключи для разных субъектов могут иметь одинаковые значения **CKA_ID**, и неопределенности не возникнет.)

В целях совместимости значения атрибутов имя субъекта (CKA_SUBJECT) и идентификатор ключа (CKA_ID) должны совпадать у сертификата и соответствующей ему ключевой пары (хотя они не обязательно должны храниться на одном и том же токене). Является необязательным условием, также как и условие уникальности идентификатора ключа для данного субъекта – в частности, приложение может оставить значение атрибута пустым.

Атрибуты **CKA_ISSUER** и **CKA_SERIAL_NUMBER** используются для совместимости с PKCS #7 и протоколом конфиденциальной почты Privacy Enhanced Mail (PEM, RFC1421). Заметим, что для сертификатов X.509 версии 3 идентификатор ключа может быть включен в дополнение сертификата. Подразумевается, что значение **CKA_ID** равно идентификатору ключа в таком сертификате, хотя это требование является необязательным.

Атрибут **CKA_URL** делает возможным хранение URL, где можно найти сертификат. Хранение URL вместо сертификата часто используется в мобильной среде.

Атрибуты **CKA_HASH_OF_SUBJECT_PUBLIC_KEY** и **CKA_HASH_OF_ISSUER_PUBLIC_KEY** используются для хранения хешей открытых ключей владельца и издателя сертификата. Они особенно важны в случае, если доступен только URL для связи сертификата с закрытым ключом, и при поиске сертификата издателя.

Атрибут **CKA_JAVA_MIDP_SECURITY_DOMAIN** связывает сертификат с доменом безопасности Java MIDP.

Ниже приводится пример шаблона для создания объекта сертификата X.509.

Шаблон создания объекта сертификата X.509

```
CK_OBJECT_CLASS your_class = CKO_CERTIFICATE;
CK_CERTIFICATE_TYPE certType = CKC_X_509;
CK_UTF8CHAR label[] = "A certificate object";
CK_BYTE subject[] = {...};
CK_BYTE id[] = {123};
CK_BYTE certificate[] = {...};
CK_BBOOL IsTrue = CK_TRUE;
CK_ATTRIBUTE template[] = {
  {CKA_CLASS, &your_class, sizeof(your_class)},
  {CKA_CERTIFICATE_TYPE, &certType, sizeof(certType)};
  {CKA_TOKEN, &IsTrue, sizeof(IsTrue)},
  {CKA_LABEL, label, sizeof(label)-1},
  {CKA_SUBJECT, subject, sizeof(subject)},
  {CKA_ID, id, sizeof(id)},
  {CKA_VALUE, certificate, sizeof(certificate)}
};
```

к содержанию ↑

Объекты сертификата атрибутов X.509

Объекты сертификата атрибутов X.509 (тип сертификата **CKC_X_509_ATTR_CERT**) содержат сертификаты атрибутов X.509. Следующая таблица определяет атрибуты объектов сертификата атрибута X.509 в дополнение к общим атрибутам, определенным для этого класса объектов .

Таблица 2.8. Атрибуты объектов сертификата атрибутов X.509

Атрибут	Тип данных	Значение
CKA_OWNER[1]	Byte Array	Поле владельца сертификата атрибутов в DER-кодировке Отличается от атрибута CKA_SUBJECT сертификатов CKC_X_509, поскольку отличается синтаксис ASN.1 и кодировка
CKA_AC_ISSUER	Byte Array	Поле издателя сертификата атрибутов в DER-кодировке. Отличается от атрибута CKA_ISSUER сертификатов CKC_X_509, поскольку отличается синтаксис ASN.1 и кодировка (по умолчанию пусто)
CKA_SERIAL_NUMBER	Byte Array	Серийный номер сертификата в DER-кодировке (по умолчанию пусто)
CKA_ATTR_TYPES	Byte Array	Последовательность значений идентификаторов объектов, соответствующих типам содержащихся атрибутов в сертификате в BER-кодировке. Поле позволяет осуществлять поиск заданного атрибута без анализа и разбора самого сертификата (по умолчанию пусто)

CKA_VALUE[1]	Byte Array	Сертификат в BER-кодировке

[1] должен быть задан при создании объекта

Только атрибуты **CKA_AC_ISSUER**, **CKA_SERIAL_NUMBER** и **CKA_ATTR_TYPES** могут быть изменены после создания объекта.

Ниже приводится пример шаблона для создания объекта сертификата атрибута X.509.

Шаблон для создания объекта сертификата атрибута X.509

```
CK_OBJECT_CLASS your_class = CKO_CERTIFICATE;
CK_CERTIFICATE_TYPE certType = CKC_X_509_ATTR_CERT;
CK_UTF8CHAR label[] = "An attribute certificate object";
CK_BYTE owner[] = {...};
CK_BYTE certificate[] = {...};
CK_BBOOL IsTrue = CK_TRUE;
CK_ATTRIBUTE template[] = {
  {CKA_CLASS, &your_class, sizeof(your_class)},
  {CKA_CERTIFICATE_TYPE, &certType, sizeof(certType)};
  {CKA_TOKEN, &IsTrue , sizeof(IsTrue )},
  {CKA_LABEL, label, sizeof(label)-1},
  {CKA_OWNER, owner, sizeof(owner)},
  {CKA_VALUE, certificate, sizeof(certificate)}
};
```

к содержанию ↑