Сценарии использования

Использование ключей на Рутокене

`rtengine` позволяет использовать ключи, расположенные на токене.

Ключевая пара идентифицируется с помощью pkcs11 uri.

Возможные компоненты идентификатора пути:

- `manufacturer:` ID производителя токена
- `model:` модель токена
- `serial:` серийный номер токена
- `token:` метка токена(поле `"label"`)
- `object:` имя объекта(`CKA_LABEL`)
- `id:` идентификатор объекта (`CKA_ID`)

Пример идентификатора ключевой пары на токене:

```
pkcs11:manufacturer=Aktiv%20Co.;model=Rutoken%20ECP;serial=2adc8d87;object=my%20label;id=%aa%bb%cc%dd?pin-value=12345678
```

Если подключен только один Рутокен с единственной ключевой парой:

```
pkcs11:model=Rutoken%20ECP
```

В зависимости от операции будет выбран открытый или закрытый ключ соответственно. Оба ключа пары должны иметь одинаковый идентификатор объекта и/или имя объекта.

Генерация ключевой пары ГОСТ в файл

`openssl genpkey -algorithm gost2012_256 -pkeyopt paramset:A -out seckey.pem`

Укажите алгоритм, используя опцию -algorithm.

Укажите параметры эллиптической кривой (парамсет) с помощью -pkeyopt.

Поддерживаются следующие значения алгоритмов и соответствующих им парамсетов:

- `gost2001: A,B,C,XA,XB`
- `gost2012_256: A,B,C,XA,XB`
- `gost2012_512: A,B`

Генерация ключевой пары ГОСТ на Рутокене

Через OpenSSL пока не поддерживается. Для генерации на Рутокене используйте pkcs11-tool из состава OpenSC.

- **`-2001`**

`pkcs11-tool.exe --module rtPKCS11ECP.dll --login --pin 12345678 --keypairgen  --key-type GOSTR3410:A --id 3132 --usage-derive`

`pin`: PIN-код Рутокена
`id`: идентификатор объекта (CKA_ID) в виде двузначных номеров символов в hex из таблицы ASCII.
Чтобы использовать этот id через OpenSSL надо использовать символы, соответствующие этим кодам. Например: для '`--id 3132`' в OpenSSL надо указывать "`pkcs11:id=12`".
Есть удобный онлайн-сервис конвертации строки в ASCII-коды

`usage-derive`: флаг указывающей, что на сгенерированном ключе можно вырабатывать общий симметричный ключ, который может использоваться, например, для расшифрования cms сообщений.

Если вы не планируйте шифровать сообщения на генерируемой ключевой паре, то этот флаг можно убрать.

`GOSTR3410:A` : 'A' - парамсет, может быть так же B или C

`-2012`

Собирайте ветку pkcs11-tool с поддержкой ГОСТ-2012, или используйте релиз OpenSC 0.20.0 или новее

```
pkcs11-tool.exe --module rtPKCS11ECP.dll --login --pin 12345678 --keypairgen --key-type GOSTR3410-2012-256:B --id 3132 --usage-derive
```

`pin`: PIN-код Рутокена

`id`: идентификатор объекта (CKA_ID) в виде двузначных номеров символов в hex из таблицы ASCII.
Чтобы использовать этот id через OpenSSL надо использовать символы, соответствующие этим кодам.
Например: для '`--id 3132`' в OpenSSL надо указывать "`pkcs11:id=12`".
Для удобства, можно воспользоваться онлайн-сервисом конвертации ACSII-кодов в строку.

`usage-derive`: флаг указывающей, что на сгенерированном ключе можно вырабатывать общий симметричный ключ, который может использоваться, например, для расшифрования cms сообщений.

Если вы не планируйте шифровать сообщения на генерируемой ключевой паре, то этот флаг можно убрать.

`GOSTR3410-2012-512:A` : 'A' - парамсет, может быть так же B или C,

`GOSTR3410-2012-256:B` : 'B' - парамсет, может быть так же C или D

или

используйте веб-сервис "Центр регистрации Рутокен"

:

```
pkcs11-tool.exe --module rtPKCS11ECP.dll -Ol
```

Формирование запроса PKCS#10

:

```
openssl req -utf8 -new -key seckey.pem -out req.csr
```

:

```
openssl req -utf8 -new -keyform engine -key "pkcs11:your_pkcs11_uri" -engine rtengine -out req.csr
```

Формат параметра `-key` описан в разделе "Использование ключей на Рутокене"

В процессе работы команда попросит ввести PIN-код. После этого потребуется указать данные для сертификата:

- `State or Province []: Moscow`
- `Locality []:  RU`
- `Organization Name []: Aktiv Company`
- `Organizational Unit Name []: development`
- `Common Name []: tester`
- `Email []: tester@rutoken.ru`

Набор вводимой информации при формировании запроса определяется конфигурационным файлом `openssl.cnf`.

Выпуск самоподписанного сертификата по запросу

:

```
openssl req -utf8 -x509 -key seckey.pem -out cert.cer
```

:

```
openssl req -utf8 -x509 -keyform engine -key "pkcs11:your_pkcs11_uri" -engine rtengine -out cert.cer
```

Создание подписи в формате CMS

Для создания CMS подписи необходимо иметь сертификат. В тестовых целях в папке `sdk\openssl\samples\tool\` предоставлены настройки удостоверяющего центра OpenSSL, который позволяет выпускать сертификаты.

Скопируйте папку `sdk\openssl\samples\tool\demoCA` и конфигурационный файл `openssl.cnf` в папку с OpenSSL и выполните:

```
openssl ca -batch -in req.csr -out cert.cer
```

CMS :

```
openssl cms -sign -binary -nosmimecap -in data_to_sign -out signed_cms -outform PEM -inkey seckey.pem -signer cert.cer
```

:

```
openssl cms -sign -binary -nosmimecap -in data_to_sign -out signed_cms -outform PEM -keyform engine -inkey "pkcs11:your_pkcs11_uri" -engine rtengine -signer cert.cer
```

Используя `-nodetach` подписываемые данные включаются в состав CMS пакета — присоединенная подпись. Без этой опции подпись будет «отсоединенной».

Используя `-nocerts` сертификат подписанта не включается в состав CMS пакета.

Проверка подписи в формате CMS

```
openssl cms -verify -binary -in signed_cms -inform PEM -out verified_data -CAfile demoCA/cacert.pem -content data_to_sign
```

Файл, указанный в `-CAfile`, является доверенным сертификатом удостоверяющего

центра и используется для проверки сертификата подписанта.

В опцию `-content` передается файл с подписанными данными, если он не был включен в состав CMS пакета.

Если сертификат подписанта не был включен в CMS пакет (отсоединенная подпись), он указывается в опции -certfile.

«Сырая» подпись данных

:

```
openssl dgst -sign seckey.pem -out signature data_to_sign
```

:

```
openssl dgst -keyform engine -sign "pkcs11:your_pkcs11_uri" -engine rtengine -out signature data_to_sign
```

Алгоритм хеша будет зависеть от алгоритма ключа.

Проверка «сырой подписи»

:

```
openssl pkey -in seckey.pem -pubout -out pubkey.pem
```

:

```
openssl dgst -verify pubkey.pem -signature signature data_to_sign
```

:

```
openssl dgst -keyform engine -verify "pkcs11:your_pkcs11_uri" -engine rtengine -signature signature test_data
```

Шифрование в формате CMS

Шифрование на ключах с Рутокена

При расшифровании сообщения вырабатывается общий симметричный ключ, который непосредственно и используется при расшифровке. Рутокен позволяет генерировать такой общий ключ только на не извлекаемых закрытых ключах с опцией 'derive' в поле key usage. Для того чтобы указать эту опцию, при генерации ключа используйте флаг --usage-derive. Например:

```
pkcs11-tool.exe --module rtPKCS11ECP.dll --login --pin 12345678 --keypairgen --key-type GOSTR3410-2012-256:B --id 3132 --usage-derive
```

```
openssl cms -encrypt -binary -<gost28147-xxx-alg> -in test_data -out encrypted_cms -outform PEM respondent.cer
```

`respondent.cer:` сертификат адресата, для которого шифруется сообщение.
Где **gost28147-xxx-alg:**

`gost28147-paramset_a-cfb ( rtengine 0.7):` алгоритм, работает в режиме гаммирования с обратной связью с набором параметров A.

`gost28147-cfb ( rtengine 0.7):` алгоритм, работает в режиме гаммирования с обратной связью с набором параметров Z.

Расшифрование на стороне адресата:

:

```
openssl cms -decrypt -binary -in encrypted_cms -inform PEM -recip respondent.cer -inkey seckey.pem -out decrypted_cms_data
```

:

```
openssl cms -decrypt -binary -in encrypted_cms -inform PEM -recip respondent.cer -keyform engine -inkey "pkcs11:your_pkcs11_uri" -engine rtengine -out decrypted_cms_data
```

Запуск SSL/TLS сервера/клиента

, :

```
openssl s_server -key demoCA/private/cakey.pem -cert demoCA/cacert.pem -Verify 7 -CAfile demoCA/cacert.pem -accept 44330 -WWW -purpose any -4
```

, :

```
openssl s_server -keyform engine -key "pkcs11:server_key_pkcs11_uri" -engine rtengine -cert demoCA/cacert.pem -Verify 7 -CAfile demoCA/cacert.pem -accept 44330 -WWW -purpose any -4
```

, :

```
openssl s_client -host 127.0.0.1 -port 44330 -cert cert.cer -key seckey.pem
```

, :

```
openssl s_client -host 127.0.0.1 -port 44330 -cert cert.cer -keyform engine -key "pkcs11:client_key_pkcs11_uri" -engine rtengine
```