

Настройка удаленного доступа к серверу по OpenSSH с устройством Рутокен

[Введение](#)

[Стенд](#)

[Порядок действий для библиотеки librtpkcs11ecp.so](#)

0. Проверка модели устройства

1. Генерация ключей

2. Настройка сервера

3. Настройка *nix-клиента

4. Настройка Windows-клиента

[Порядок действий для библиотеки opensc-pkcs11.so](#)

1. Настройка сервера

2. Настройка клиента

[Дополнительная информация](#)

Введение

Здесь приводится инструкция по настройке доступа к удаленному серверу с помощью OpenSSH и Рутокен ЭЦП. Рутокен работает как с библиотекой **librtpkcs11ecp.so**, так и библиотекой **pkcs11.so** из состава OpenSC.

Стенд

- Сервер Ubuntu x86,
- Клиент Ubuntu x86,
- Клиент Windows 10;
- Рутокен ЭЦП, отформатированный через Панель управления Рутокен.

Порядок действий для библиотеки **librtpkcs11ecp.so**

0. Проверка модели устройства

1. Подключите USB-токен к компьютеру.
2. Для определения названия модели USB-токена откройте **Терминал** и введите команду:

```
$ lsusb
```

В результате в окне Терминала отобразится название модели USB-токена:

```
[dmitrieva@localhost ~]$ lsusb
Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
Bus 002 Device 004: ID 0a89:0030 Aktiv Rutoken ECP
Bus 002 Device 003: ID 0e0f:0002 VMware, Inc. Virtual USB Hub
Bus 002 Device 002: ID 0e0f:0003 VMware, Inc. Virtual Mouse
Bus 002 Device 001: ID 1d6b:0001 Linux Foundation 1.1 root hub
```

Убедитесь, что используете: **Aktiv Rutoken ECP**

1. Генерация ключей

Первый вариант

На сервере или любой клиентской *nix машине выполняем следующие действия:

1.1 Устанавливаем необходимые для работы с Рутокен пакеты:

```
$ sudo apt-get install opensc
```

1.2 Устанавливаем библиотеку librtpkcs11ecp.so (<http://www.rutoken.ru/support/download/pkcs/>)

1.3 Генерируем ключевую пару на Рутокен:

```
$ pkcs11-tool --module /usr/lib/librtpkcs11ecp.so --keypairgen --key-type rsa:2048 -l --id 45
```

1.4 Конвертируем в формат ssh открытый ключ на Рутокен:

```
$ ssh-keygen -D /usr/lib/librtpkcs11ecp.so -I 0:45 >> key.pub
```

Здесь пара 0:45 - это <слот>:<id>.

1.5 Дополнительно для использования SSH-клиента [PuttySC](#) на Windows

1.5.1 Выписываем сертификат для сгенерированной ключевой пары:

```
$ sudo apt-get install libengine-pkcs11-openssl
$ openssl
OpenSSL> engine dynamic -pre SO_PATH:/usr/lib/engines-1.1/pkcs11.so -pre ID:pkcs11 -pre LIST_ADD:1 -pre LOAD -
pre MODULE_PATH:/usr/lib/librtpkcs11ecp.so
OpenSSL> req -engine pkcs11 -new -key 0:45 -keyform engine -x509 -out cert.cert -text -days 365 -text
OpenSSL> exit
```

1.5.2 Конвертируем сертификат в DER-формат:

```
$ openssl x509 -in cert.cert -out cert.der -outform der
```

1.5.3 Импортируем сертификат на Рутокен:

```
$ pkcs11-tool --module /usr/lib/librtpkcs11ecp.so -l -y cert -w cert.der --id 45 --label Rutoken1
```

Второй вариант

На сервере или любой клиентской *nix машине выполняем следующие действия:

1.1 Устанавливаем openssl.

1.2 Библиотеку librtpkcs11ecp.so помещаем в директорию /usr/lib/

1.3 Устанавливаем openssl-client и openssl:

```
$ sudo apt-get install openssl-client openssl
```

1.4 Генерируем ключевую пару:

```
$ openssl genrsa -out keys.pem 2048
```

1.5 Создаем самоподписанный сертификат:

```
$ openssl req -new -key keys.pem -out cert.csr
$ openssl x509 -req -days 700 -in cert.csr -signkey keys.pem -out cert.cert
```

1.6 Конвертируем ключи и сертификат в DER-формат:

```
$ openssl rsa -inform PEM -in keys.pem -out keys.der -outform DER
$ openssl x509 -in cert.cert -out cert.der -outform der
```

1.7 Импортируем ключи и сертификат в DER-формате на Рутокен:

```
$ pkcs11-tool --module /usr/lib/librtpkcs11ecp.so -l -y privkey -w keys.der --id 10 --label Rutoken1  
$ pkcs11-tool --module /usr/lib/librtpkcs11ecp.so -l -y cert -w cert.der --id 10 --label Rutoken1
```

1.8 Конвертируем открытый ключ на Рутокен в формат ssh:

```
$ ssh-keygen -D /usr/lib/librtpkcs11ecp.so -I 0:10 >> key.pub
```

Здесь пара 0:10 - это <слот>:<id>.

2. Настройка сервера

2.1 Устанавливаем openssh-server:

```
$ sudo apt-get install opensc openssh-server
```

2.2 Копируем на сервер содержимое полученного на шаге 1.4 (1.8 второй вариант) файла key.pub в файл ~/.ssh/authorized_keys (если такого файла нет, нужно его создать). Для файла с приватными ключами измените права доступа с помощью команды:

```
$ chmod 0600 ~/.ssh/authorized_keys
```

3. Настройка *nix-клиента

3.1 Устанавливаем opensc и openssh-client:

```
$ sudo apt-get install openssh-client
```

3.2 Устанавливаем библиотеку librtpkcs11ecp.so (<http://www.rutoken.ru/support/download/pkcs/>)

3.3 Вызываем

```
ssh-agent -p /opt/aktivco/rutokenecp/amd64/*
```

3.4 Подключаемся к серверу:

```
ssh -I /usr/lib/librtpkcs11ecp.so <username>@<server>
```

4. Настройка Windows-клиента

4.1 Устанавливаем драйверы Рутокен <http://www.rutoken.ru/support/download/drivers-for-windows/>.

4.2 Начиная с Windows 10, можно использовать SSH через Командную строку или PowerShell, с указанием пути к нашей библиотеки rtpkcs11ecp.dll.

```
ssh -I "C:\Windows\System32\rtpkcs11ecp.dll" <username>@<server>
```

Порядок действий для библиотеки **opensc-pkcs11.so**

1. Настройка сервера

1.1 Устанавливаем openssh-server:

```
$ sudo apt-get install openssh-server
```

1.2 Форматируем Рутокен:

```
$ pkcs15-init --erase-card -p rutoken_ecp  
$ pkcs15-init --create-pkcs15 --so-pin "87654321" --so-puk ""  
$ pkcs15-init --store-pin --label "User PIN" --auth-id 02 --pin "12345678" --puk "" --so-pin "87654321" --  
finalize
```

1.3 Генерируем ключевую пару на Рутокен:

```
$ pkcs15-init -G rsa/2048 --auth-id 02 --id 42
```

1.4 Конвертируем открытый ключ в формат ssh:

```
$ pkcs15-tool --read-ssh-key 42
```

Здесь 42 – это id ключа.

1.5 Полученный результат копируем в файл ~/.ssh/authorized_keys на сервере.

2. Настройка клиента

2.1 Устанавливаем opensc и openssh-client:

```
$ sudo apt-get install opensc openssh-client
```

2.2 Вызываем

```
ssh-agent -p < ( symlink ) opensc-pkcs11.so>/*
```

2.3 Подключаемся к серверу:

```
$ ssh -I /usr/lib/opensc-pkcs11.so <username>@<server>
```

В случае подключения с использованием конфигурационного файла следует вместо параметра IdentityFile добавить SmartcardDevice со значением /usr/lib/librtpkcs11ecp.so.

Дополнительная информация

1. [Using OpenSSH with smartcards](#)