

# Аутентификация в AlterOS при помощи RSA ключей на Рутокен ЭЦП

- [Создание ключей и сертификатов](#)
- [Добавление сертификата в список доверенных](#)
- [Настройка pam\\_pkcs11](#)
- [Регистрация модуля для аутентификации в системе](#)

Подключите устройств семейства Рутокен ЭЦП к компьютеру.

Перед началом работы, установите следующие пакеты:

```
sudo yum update
sudo yum install ccid opensc pam_pkcs11 gdm-plugin-smartcard p11-kit openssl-pkcs11
# -
sudo pcsd
```

Загрузите модуль [librtpkcs11ecp.so](#) и установите:

```
sudo rpm -i librtpkcs11ecp-2.3.0.0-1.x86_64.rpm
```

## Создание ключей и сертификатов

Вы можете пропустить данный раздел, если у вас уже имеются необходимые RSA ключи.

Если ключей нет, ниже команда для их созданию:

```
pkcs11-tool --module /usr/lib64/librtpkcs11ecp.so --keypairgen --key-type rsa:2048 -l --id 45
```

Параметр id задает идентификатор ключевой пары.

Теперь нужно получить сертификат:

- создайте самоподписанный сертификат:

```
openssl

OpenSSL> openssl engine dynamic -pre SO_PATH:/lib64/openssl/engines/libpkcs11.so -pre ID:pkcs11 -pre LIST_ADD:1
-pre LOAD -pre MODULE_PATH:/usr/lib64/librtpkcs11ecp.so

OpenSSL> req -engine pkcs11 -new -key 0:45 -keyform engine -x509 -out cert.crt -outform DER
```

- или создайте запрос на сертификат для передачи его в УЦ:

```
openssl
```

```
OpenSSL> openssl engine dynamic -pre SO_PATH:/lib64/openssl/engines/libpkcs11.so -pre ID:pkcs11 -pre LIST_ADD:1  
-pre LOAD -pre MODULE_PATH:/usr/lib64/librtpkcs11lecp.so
```

```
OpenSSL> req -engine pkcs11 -new -key 0:45 -keyform engine -out request.req
```

Сохраните сертификат на токене:

```
pkcs11-tool --module /usr/lib64/librtpkcs11lecp.so -l -y cert -w cert.crt --id 45
```

Проверьте, что токен подключен и на нем сохранены сертификаты и ключи.

## Добавление сертификата в список доверенных

Создайте базу данных доверенных сертификатов

```
sudo mkdir /etc/pam_pkcs11/nssdb  
sudo chmod 0644 /etc/pam_pkcs11/nssdb  
sudo certutil -d /etc/pam_pkcs11/nssdb -N #  
sudo modutil -dbdir /etc/pam_pkcs11/nssdb/ -add p11-kit-trust -libfile /usr/lib64/pkcs11/p11-kit-trust.so
```

Выгрузите ваш сертификат с токена (если вы пользовались для получения сертификата вышеописанной инструкцией , то ID = 45):

```
pkcs11-tool --module=/usr/lib64/librtpkcs11lecp.so -l -r -y cert -d <ID> -o cert.crt
```

Добавьте сертификат в доверенные:

```
sudo cp cert.crt /etc/pki/ca-trust/source/anchors/ #  
sudo update-ca-trust force-enable  
sudo update-ca-trust extract #
```

## Настройка pam\_pkcs11

Создайте (например, на рабочем столе) текстовый файл pam\_pkcs11.conf со следующим содержимым:

```

pam_pkcs11 {
    nullok = false;
    debug = false;
    use_first_pass = false;
    use_authtok = false;
    card_only = false;
    wait_for_card = false;
    use_pkcs11_module = rutokenecp;

    # Aktiv Rutoken ECP
    pkcs11_module rutokenecp {
        module = /usr/lib64/librtpkcs11ecp.so;
        slot_num = 0;
        support_thread = true;
        ca_dir = /etc/pam_pkcs11/cacerts;
        crl_dir = /etc/pam_pkcs11/crls;
        cert_policy = signature;
    }

    use_mappers = digest;

    mapper_search_path = /usr/lib64/pam_pkcs11;

    mapper digest {
        debug = false;
        module = internal;
        algorithm = "sha1";
        mapfile = file:///etc/pam_pkcs11/digest_mapping;
    }
}

```

Поместите файл в каталог /etc/pam\_pkcs11/:

```

cd /etc/pam_pkcs11/

sudo mv pam_pkcs11.conf pam_pkcs11.conf.default #

sudo mkdir cacerts crls

sudo cp /path/to/your/pam_pkcs11.conf /etc/pam_pkcs11/

```

## Регистрация модуля для аутентификации в системе

Подключите модуль к системе авторизации PAM:

```

sudo vim /etc/pam.d/system-auth
#
sudo vim /etc/pam.d/password-auth

```

Перед первым использованием модуля pam\_unix добавьте туда строку со следующим содержимым:

```

auth    sufficient          pam_pkcs11.so pkcs11_module=/usr/lib64/librtpkcs11ecp.so

```

Сохраните файл и узнайте поля вашего сертификата с помощью следующей команды:

```
sudo pkcs11_inspect
```

В результате отобразится сообщение:

```
[tester@localhost]$ sudo pkcs11_inspect
PIN for token:
DEBUG:subject_mapper.c:116: Subject mapper started. debug: 1, mapfile: file:///etc/pam_pkcs11/subject_mapping,
icase: 0
Printing data for mapper subject:
E=tester@mail.ru,CN=tester,OU=test,O=test,L=msk,ST=msk,C=ru
```

Скопируйте строчку с описанием сертификата в файл /etc/pam\_pkcs11/digest\_mapping в формате:

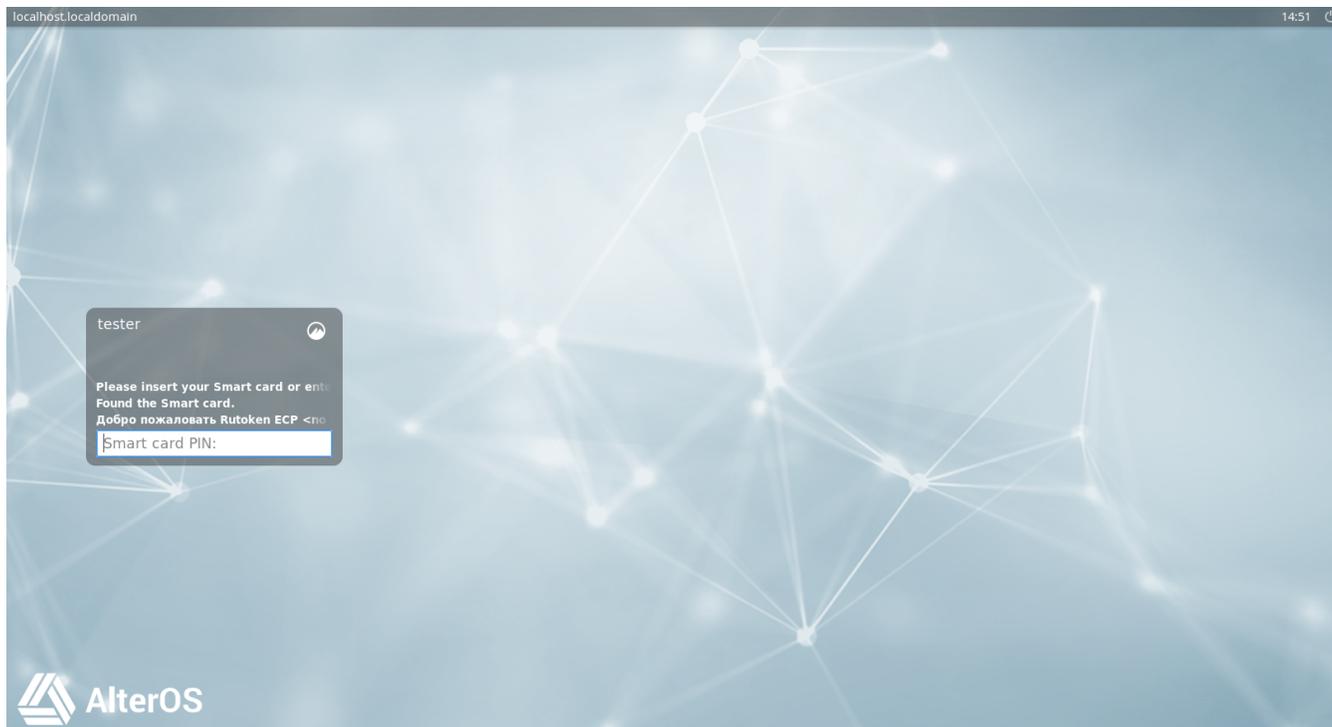
```
< pkcs11_inspect> -> <_>
```

```
[tester@localhost]$ cat /etc/pam_pkcs11/subject_mapping
E=tester@mail.ru,CN=tester,OU=test,O=test,L=msk,ST=msk,C=ru -> tester
```

Попробуйте аугентифицироваться:

```
[tester@localhost]$ su tester
Please insert your Smart card or enter your username.
Found the Smart card.
  Rutoken ECP <no label>!
Smart card PIN:
```

В окне экрана приветствия аналогично:



## Настройка автоблокировки

В состав пакета libram-pkcs11 входит утилита pkcs11\_eventmgr, которая позволяет выполнять различные действия при возникновении событий PKCS#11.

Для настройки pkcs11\_eventmgr служит файл конфигурации - /etc/pam\_pkcs11/pkcs11\_eventmgr.conf

Пример файла конфигурации представлен ниже:

```
pkcs11_eventmgr
{
    #
    daemon = true;

    #
    debug = false;

    #
    polling_time = 1;

    # -
    # - 0
    expire_time = 0;

    # pkcs11
    pkcs11_module = /usr/lib64/librtpkcs11ecp.so;

    #
    # :
    event card_insert {
        # ( )
        on_error = ignore ;

        action = "/bin/false";
    }

    #
    event card_remove {
        on_error = ignore;

        #
        action = "cinnamon-screensaver-command --lock";
    }

    #
    event expire_time {
        # ( )
        on_error = ignore;

        action = "/bin/false";
    }
}
```

После этого добавьте приложение pkcs11\_eventmgr в автозагрузку и перезагрузите компьютер.

Для этого создайте файл /etc/xdg/autostart/smartcard-screensaver.desktop

```
[Desktop Entry]
Type=Application
Name=Smart Card Screensaver
Comment=Application to lock screen on smart card removal.
Exec=/usr/bin/pkcs11_eventmgr daemon
```