Работа с КриптоПро на linux сервере

Ссылки

- Установка КриптоПро на Debian/Ubuntu
- Документация [http://www.cryptopro.ru/sites/default/files/products/cryptcp/3-33/CryptCP.pdf]

Лицензия

Просмотр лицензии:

```
cpconfig -license -view
```

Для установки другой лицензии (под root):

```
cpconfig -license -set <серийный_номер>
```

Корневые сертификаты

Просмотр корневых сертификатов

```
certmgr -list -store uroot
```

В более старых версиях вместо `uroot` следует использовать `root`:

```
certmgr -list -store root
```

Добавление корневых сертификатов (под root) из файла cacer.p7b [http://cpca.cryptopro.ru/cacer.p7b]

```
sudo certmgr -inst -all -store uroot -file cacer.p7b
```

Необходимо последовательно добавить все сертификаты

Сертификаты

Список установленных сертификатов

`certmgr -list`, например:

```
1-------
Issuer            : E=support@cryptopro.ru, C=RU, L=Moscow, O=CRYPTO-PRO LLC, CN=CRYPTO-PRO Test Center 2
Subject           : CN=test2
Serial            : 0x120007E4E683979B734018897B00000007E4E6
SHA1 Hash         : 0x71b59d165ab5ea39e4cd73384f8e7d1e0c965e81
Not valid before  : 07/09/2015  10:41:18 UTC
Not valid after   : 07/12/2015  10:51:18 UTC
PrivateKey Link   : Yes. Container  : HDIMAGE\\test2.000\F9C9
2-------
Issuer            : E=support@cryptopro.ru, C=RU, L=Moscow, O=CRYPTO-PRO LLC, CN=CRYPTO-PRO Test Center 2
Subject           : CN=webservertest
Serial            : 0x120007E47F1FD9AE0EDE78616600000007E47F
SHA1 Hash         : 0x255c249150efe3e48f1abb3bc1928fc8f99980c4
Not valid before  : 07/09/2015  09:56:10 UTC
Not valid after   : 07/12/2015  10:06:10 UTC
PrivateKey Link   : Yes. Container  : HDIMAGE\\webserve.001\2608
```

Добавление реального сертификата

Добавить только сертификат (только проверка ЭЦП):

```
certmgr -inst -file cert.cer
```

Добавление реального сертификата с привязкой к закрытому ключу и возможностью подписывать документы

Закрытый ключ состоит из шести key-файлов:

```
header.key
masks2.key
masks.key
name.key
primary2.key
primary.key
```

Способ с дискетой или флешкой

Скопировать в корень дискеты или флэшки сертификат и приватный ключ (из каталога `999996.000`, `999996` - название (alias) контейнера):

```
cp -R /path/to/key/999996.000 /media/flashdrive/
cp /path/to/cert/client.cer /media/flashdrive/
```

Выполнить команду по копированию ключа с флэшки на диск, ключ попадет в пользовательское хранилище `My`.

Необходимо выполнять под пользователем, который будет использовать данный контейнер для подписи.

gate@example.com - то, что прописано в поле E сертификата (можно посмотреть командой `keytool --printcert -file /path/to/cert/client.cer`):

```
csptest -keycopy -src '\\.\FLASH\gate@example.com' -dest '\\.\HDIMAGE\999996'
```

С жесткого диска

«Ручной способ».

Скопировать приватный ключ в хранилище (контейнер), где `<username>` - имя пользователя linux:

```
cp -R /path/to/key/999996.000 /var/opt/cprocsp/keys/<username>/
```

Поставить «минимальные» права:

```
chmod 600 /var/opt/cprocsp/keys/<username>/999996.000/*
```

Узнать реальное название контейнера:

```
csptest -keyset -enum_cont -verifycontext -fqcn
```

Ассоциировать сертификат с контейнером, сертификат попадет в пользовательское хранилище `My`:

```
certmgr -inst -file /path/to/file/client.cer -cont '\\.\HDIMAGE\999996'
```

Если следующая ошибка, нужно узнать реальное название контейнера (см. выше):

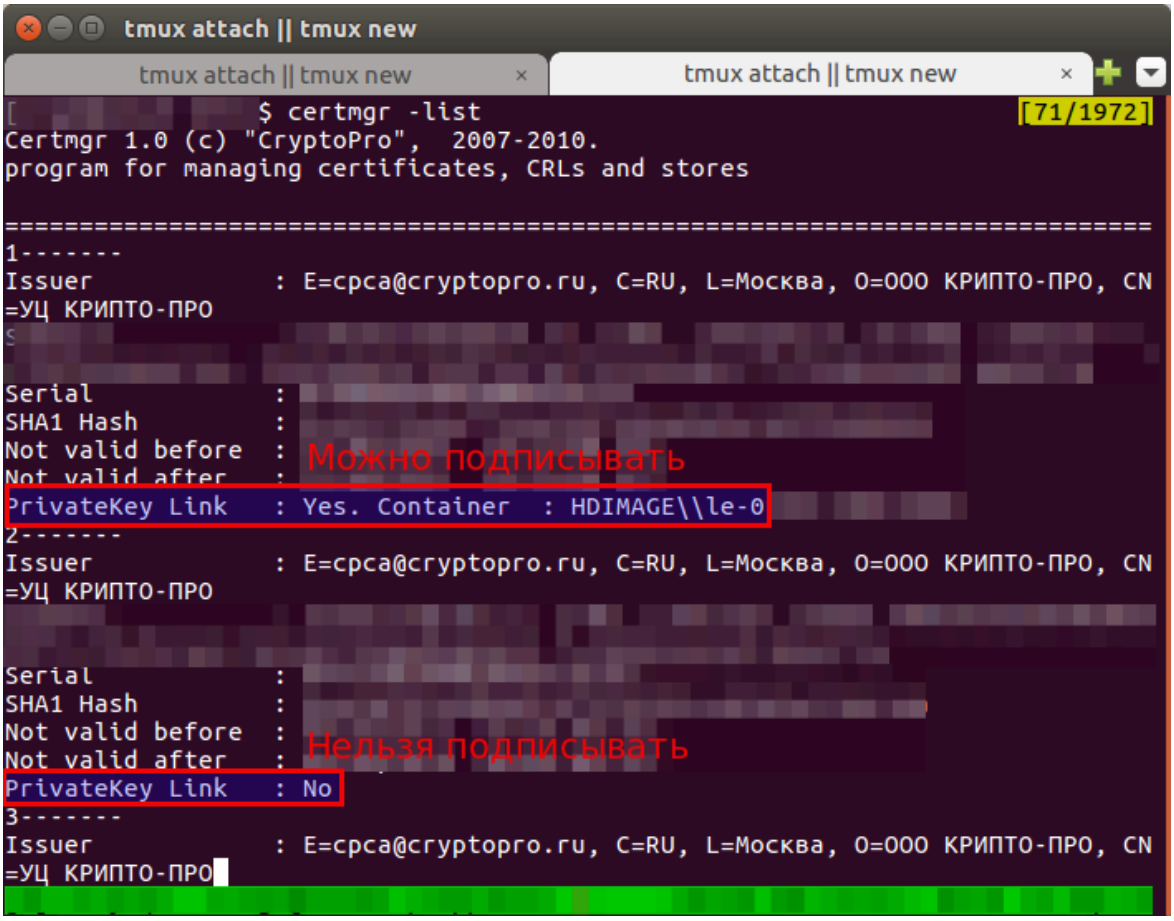
```
Failed to open container \\.\HDIMAGE\<container>
[ErrorCode: 0x00000002]
```

Установить сертификат УЦ из-под пользователя root командой:

```
certmgr -inst -store uroot -file /path/to/file/CA.cer
```

Проверка успешности установки закрытого ключа

```
certmgr --list
```



Если выводится `PrivateKey Link: Yes. Container: HDIMAGE\\999996.000\D7BB`, то есть и сертификат, и приватный ключ, а если выводится `PrivateKey Link: No` - связи нет, и использовать такой контейнер для подписи не удастся.

Источник [http://grigory-panov.blogspot.ru/2012/06/cryptopro.html]

Добавление тестового сертификата

Добавление работает только на той же машине, и в тот же контейнер, где был сформированы следующий запрос на добавление:

```
cryptcp -creatrqst -dn 'cn=test' -cont '\\.\hdimage\test' test.csr
```

Ввести пароль на контейнер `test123`.

```
cryptcp -creatrqst -dn 'e=email@test.ru,cn="тест тест",c=rus,l="москва",o="текст тест"' -cont '\\.\hdimage\myname' myname.csr
```

Пароль `mysecurepass`

Откройте в браузере ссылку тестовый удостоверяющий центр КриптоПро [http://www.cryptopro.ru/certsrv/certrqxt.asp]

```
cryptcp -instcert -cont '\\.\hdimage\test' certnew.cer
```

Ввести пароль на контейнер. По-умолчанию: `12345678`

Удаление сертификата

```
certmgr -delete 1
```

Проверка сертификата

```
certmgr -list -f file.sig
```

Ответ:

```
1-------
Issuer            : E=cpca@cryptopro.ru, C=RU, L=Москва, O=ООО КРИПТО-ПРО, CN=УЦ КРИПТО-ПРО
Subject           : E=info@site.ru, C=RU, L=г. Москва, O="ООО ""Верес""", OU=Руководство, CN=Иванов Иван Иванович, T=Генеральный директор
Serial            : 0x75F5C86A000D00016A5F
SHA1 Hash         : 0x255c249150efe3e48f1abb3bc1928fc8f99980c4
Not valid before  : 08/12/2014  09:04:00 UTC
Not valid after   : 08/12/2019  09:14:00 UTC
PrivateKey Link   : No
```

Подписание пустого файла (размер 0) проходит успешно, но при просмотре сертификатов этого файла выдается ошибка:

```
Can't open certificate store: '/tmp/tmp.G8cd13vzfZ.sig'.
Error: No certificate found.
/dailybuilds/CSPbuild/CSP/samples/CPCrypt/Certs.cpp:312: 0x2000012D
[ErrorCode: 0x2000012d]
```

Будьте внимательны!

Просмотр всех атрибутов сертификата

В `cryptcp` нет необходимых инструментов для получения всех атрибутов сертификата. Поэтому следует использовать `openssl`, но настроив его.

Получаем SHA 1 хеши:

```
certmgr -list -f file.sig | grep 'SHA1 Hash'
```

В цикле извлекаем сертификаты:

```
cryptcp -nochain -copycert -thumbprint 255c249150efe3e48f1abb3bc1928fc8f99980c4 -f file.sig -df certificate.der -der
```

```
openssl x509 -in certificate.der -inform der -text -noout
```

Настройка `openssl` для поддержки ГОСТ [http://big-town.narod.ru/openssl.html]:

В файл `/etc/ssl/openssl.cnf`

```
openssl_conf = openssl_def # Это в начало файла
#Все что ниже в конец
[openssl_def]
engines = engine_section

[engine_section]
gost = gost_section

[gost_section]
engine_id = gost
dynamic_path = /usr/lib/ssl/engines/libgost.so # заменить реальным файлом
default_algorithms = ALL
CRYPT_PARAMS = id-Gost28147-89-CryptoPro-A-ParamSet
```

Проверка:

```
openssl ciphers | tr ":" "\n" | grep -i gost
GOST2001-GOST89-GOST89
GOST94-GOST89-GOST89
```

Экспорт сертификатов на другую машину

Закрытые ключи к сертификатам находятся тут: `/var/opt/cprocsp/keys`. Поэтому эти ключи переносятся просто: создаем архив и переносим на нужную машину в тот же каталог.

Экспорт самих сертификатов (если их 14):

```
for i in `seq 1 14`; do echo $i | certmgr -export -dest $i.cer; done
```

Переносим эти файлы на машину и смотрим, какие контейнеры есть:

```
csptest -keyset -enum_cont -verifycontext -fqcn
```

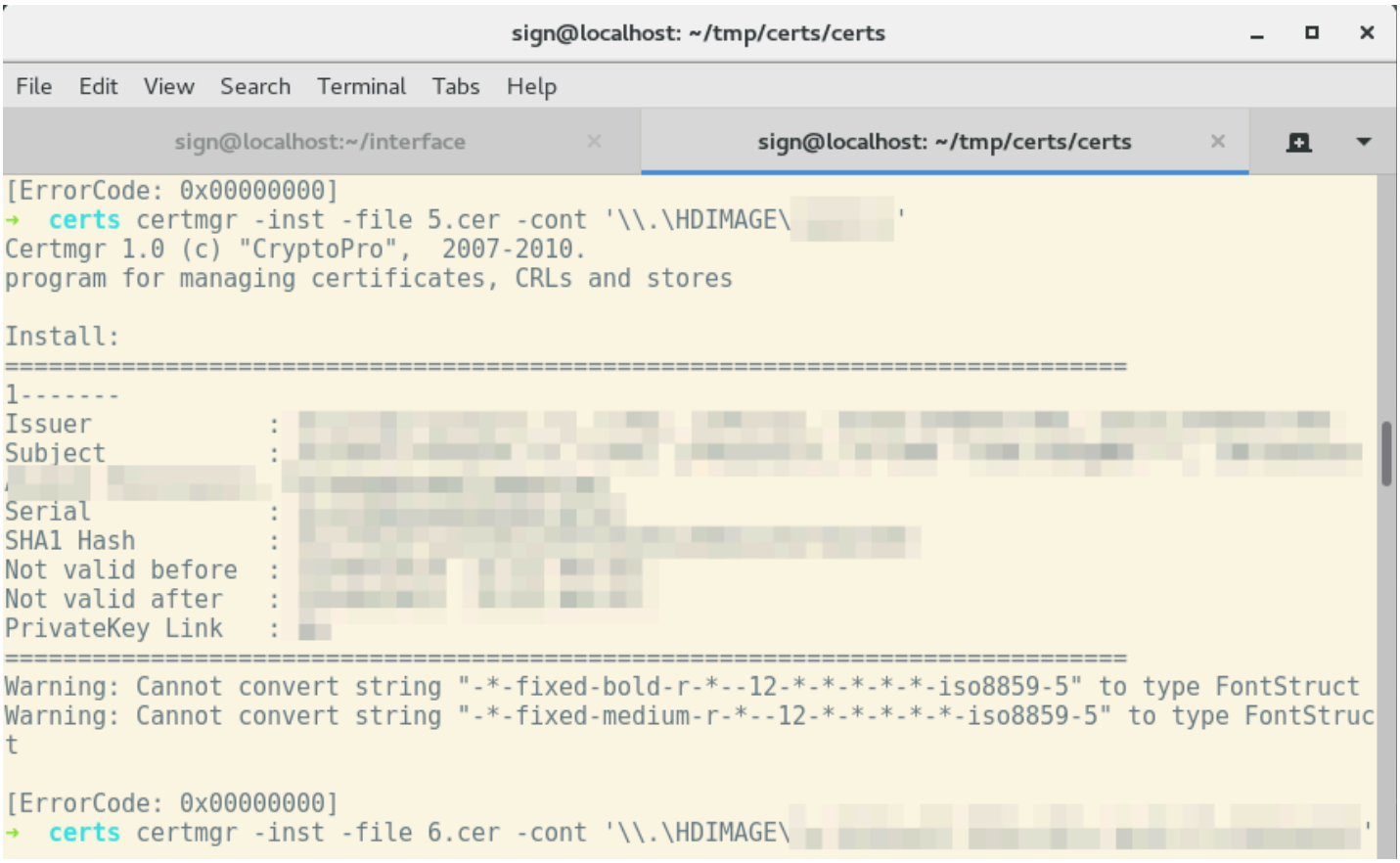
И как обычно, связываем сертификат и закрытый ключ:

```
certmgr -inst -file 1.cer -cont '\\.\HDIMAGE\container.name'
```

Если закрытый ключ и сертификат не подходят друг к другу, будет выведена ошибка:

```
Can not install certificate
Public keys in certificate and container are not identical
```

Если все успешно:



Если нет закрытого ключа, то просто ставим сертификат:

```
certmgr -inst -file 1.cer
```

Подписание документа ЭЦП

```
cryptcp -sign (КПС1) -nochain -pin pincode src.txt dest.txt.sig
```

- `nochain` - отменяет проверку цепочки сертификатов
- `pin` - пин-код
- `КПС1` - критерий поиска сертификата

Пример создания ЭЦП (по SHA1 Hash):

```
cryptcp -sign -thumbprint 255c249150efe3e48f1abb3bc1928fc8f99980c4 -nochain -pin test test.txt test.txt.sig
```

`[ReturnCode: x]`	**Описание**	**Возвращаемый код завершения в баше** `$?`
0	успешно	0
`0x8010006b`	Введен неправильный PIN	107
`0x2000012d`	Сертификат не найден	45

[ReturnCode: x]	Описание	Возвращаемый код завершения в баше $?
0x20000065	Не удалось открыть файл	101

Проверка подписи ЭЦП

Для верифицирования сертификатов нужен сертификат удостоверяющего центра и актуальный список отзыва сертификатов, либо настроенный для этого revocation provider.

Корневой сертификат УЦ [http://cpca.cryptopro.ru/cacer.p7b], список отзыва сертификата является одним из реквизитов самого сертификата.

Контрагенты когда открывают подписи в КриптоАРМ используют revocation provider, он делает проверки отзыва сертификата онлайн. Как реализована проверка в Шарепоинте не знаю. Знаю только что используется библиотека Крипто.Net

```
cryptcp -verify -nochain
```

Проверка конкретной подписи из локального хранилища по его хешу:

```
cryptcp -verify -thumbprint 255c249150efe3e48f1abb3bc1928fc8f99980c4 -nochain test.txt.sig
```

Проверить, взяв сертификат из `file1.sig`, подпись файла `file2.sig`. Практически, надо использовать один и тот же файл:

```
cryptcp -verify -norev -f file1.sig file2.sig
```

Ответ:

```
Certificates found: 2
Certificate chains are checked.
Folder './':
file.xls.sig... Signature verifying...
Signer: Старший инженер, Иванов Иван Иванович, Отдел закупок, ООО «Верес», Москва, RU, info@site.ru
 Signature's verified.
Signer: Генеральный директор, Сидоров Иван Петрович, Руководство, ООО «Кемоптика», Москва, RU, info@site.ru
 Signature's verified.
[ReturnCode: 0]
```

Результат:

[ReturnCode: x]	Текст	Описание	Возвращаемый код завершения в баше $?
0		Успешно	0
0x80091004	Invalid cryptographic message type	Неправильный формат файла	4
0x80091010	The streamed cryptographic message is not ready to return data	Пустой файл	16

Получение исходного файла

Получение исходного файла (сообщения):

```
cryptcp -verify -nochain file.sig file.txt
```

Будет ругаться на сертификат (так как не будет проверки), но подпись удалит. Вариант с проверкой:

```
cryptcp -verify -nochain -f file.sig file.sig file.txt
```

Настройка службы точного времени

```
apt-get install ntp ntpdate
```

Необходимо добавить сервера:

/etc/ntp.conf

```
# You do need to talk to an NTP server or two (or three).
#server ntp.your-provider.example
server ntp1.stratum2.ru
server ntp2.stratum2.ru
```

Ссылка [http://www.pool.ntp.org/ru/use.html]