127018, Москва, Сущёвский Вал, 18
Телефон: (495) 995 4820
https://CryptoPro.ru
E-mail: info@CryptoPro.ru

Средство Криптографической Защиты Информации

КриптоПро CSP Версия 5.0 R3 KC1 (исполнение 1-Base) Приложение командной строки cryptcp

ЖТЯИ.00101-03 93 01
Листов 31

2024 г

Содержание

Аннотация

Данный документ содержит общую информацию по использованию приложения командной строки для подписи и шифрования файлов cryptcp, предназначенного для работы с сертификатами с использованием инфраструктуры открытых ключей, шифрования/расшифрования сообщений, содержащихся в файлах, создания/проверки электронных подписей и хэширования сообщений, содержащихся в файле или группе файлов.

1 Системные требования

Приложение функционирует в программно-аппаратных средах, перечисленных в ЖТЯИ.00101-03 30 01. КриптоПро CSP. Формуляр, п. 3.2.

2 Использование программы

2.1 Запуск программы

Программа реализована в виде исполняемого файла cryptcp. Для ее запуска необходимо выполнить следующую команду:

```
[путь]cryptcp [<команда> [<опции и файлы>]]
```

путь	путь к месторасположению программы
cryptcp	имя исполняемого файла приложения
команда	одна из допустимых команд
опции	параметры команды (свои для каждой команды), начинающиеся с ''-''
файлы	имена одного или двух файлов, в зависимости от команды. Порядок файлов в командной строке относительно друг друга должен быть такой, как указано в описании команды

Примечание. К понятию файл также относятся маски файлов.

Если не указать команду, то на экран выводится список всех доступных команд с их кратким описанием. Для получения более детального описания определенной команды необходимо указать опцию -**help**.

2.2 Критерий поиска сертификатов

Критерий поиска сертификатов (далее — КПС) используется для задания сведений о субъектах, чьи сертификаты будут использоваться при выполнении команды (например, шифровании или подписи данных). Если команда такова, что КПС должен удовлетворять только один сертификат, то такой КПС будет обозначаться КПС1.

КПС задается в форме опций командной строки, которые имеют следующий синтаксис:

```
[-dn <RDN>]n раз [-issuer] [-m[<имя>]|u[<имя>]|-f <файл>]k раз [-all|-1|-q<N>]
        [-thumbprint <отпечаток>] [-nochain|-errchain [-norev|-nonet]]
```

-dn <RDN> найти сертификаты, у которых поля имен RDN совпадают со списком строк, указанным в <RDN>; при вводе нескольких -dn будет найдено большее количество сертификатов; список строк <RDN> вводится через запятую

-issuer использовать RDN издателя для поиска

-m <имя> осуществить
поиск сертификатов в хранилищах компьютера (LOCAL_MACHINE); поле <имя> (имя хранилища) по умолчанию имеет значение: "My" для создания подписи или расшифрования и "My+Addressbook" для остальных случаев

-u <имя> осуществить
поиск сертификатов в хранилищах пользователя (CURRENT_USER); поле <имя> (имя хранилища) по умолчанию имеет значение: "My" для создания подписи или расшифрования и "My+Addressbook" для остальных случаев

-f <файл> использовать сообщение или файл сертификата, название которых указывается в поле <файл>

-all использовать все найденные сертификаты (по умолчанию для КПС)

-1 должен быть найден только один сертификат, иначе вернуть ошибку (по умолчанию для КПС1)

-q<N> если найдено менее N сертификатов, вывести запрос для выбора нужного (по умолчанию N=10)

-thumbprint <отпечаток> осуществить поиск по отпечатку сертификата

-nochain не проверять цепочки найденных сертификатов

-errchain завершать выполнение с ошибкой, если хотя бы один сертификат не прошел проверку

-norev не проверять сертификаты в цепочке на предмет отозванности

-nonet использовать только кэшированные URL при построении цепочки

Примеры использования КПС можно найти в описаниях команд, использующих его.

Примечание. Если внутри опции <имя> или <RDN> присутствуют пробелы, то ее необходимо заключить в кавычки. То же относится к именам файлов и папок. Например:
Иван Иванов,a@b.c — неверно;
"**Иван Иванов,a@b.c**" — верно;
CN=Иванов,E=a@b.c — верно.

2.3 Команды шифрования/расшифрования

2.3.1 Шифрование данных

Для того, чтобы создать зашифрованное сообщение, необходимо выполнить следующую команду:

```
-encr <КПС> [-der] [-strict] [-encryptionalg <OID>]
    [-keepbadfiles] <исходный файл> <сообщение>
```

<КПС>	КПС получателей
-der	использовать формат DER вместо BASE64
-strict	использовать однозначное кодирование DER вместо BER
-encryptionalg <OID>	задать алгоритм шифрования с помощью OID
-keepbadfiles	не удалять выходной файл при ошибке
<исходный файл>	файл, содержащий исходные данные
<сообщение>	файл, который будет содержать созданное сообщение

Примечание. Для того, чтобы зашифровать данные "на себя", необходимо указать КПС своего сертификата.

Пример 1. Зашифровать содержимое файла "test.txt" в "test1.msg" (бинарный формат), используя ВСЕ сертификаты хранилища "Личные" ("My") текущего пользователя (а не локального компьютера), содержащие в поле "Субъект" ("Subject") подстроки "Иванов Петр" и "ivanov@bank.ru":

```
cryptcp -encr -dn "Иванов Петр,ivanov@bank.ru" -uMy -der test.txt test1.msg
```

Пример 2. Зашифровать содержимое файла "in.txt" в "message.enc" (формат DER), используя алгоритм ГОСТ 28147-89 и сертификат, взятый из файла "cert.p7b":

```
cryptcp -encr -f cert.p7b -der -encryptionalg 1.2.643.2.2.21 in.txt message.enc
```

2.3.2 Расшифрование данных

Для того, чтобы расшифровать сообщение, необходимо выполнить следующую команду:

```
-decr [<КПС>] [-start] [-pin <пароль>|-askpin]
  [-keepbadfiles] <сообщение> <выходной файл>
```

<КПС>	КПС получателя
-start	открыть (запустить) полученный файл
-askpin	запросить пароль ключевого контейнера из консоли
-pin <пароль>	задать пароль ключевого контейнера
-keepbadfiles	не удалять выходной файл при ошибке
<сообщение>	файл, содержащий сообщение
<выходной файл>	файл, в который будут записаны данные из сообщения

Пример 1: Расшифровать сообщение из файла "test.msg" в файл "test2.txt", используя закрытый ключ, связанный с сертификатом хранилища "Личные" ("My") текущего пользователя, содержащим в поле "Субъект" ("Subject") подстроки "Иванов Петр" и "ivanov@bank.ru", а затем открыть полученный файл:

```
cryptcp -decr -dn "Иванов Петр,ivanov@bank.ru" -start test.msg test2.txt
```

2.4 Работа с пакетами файлов

2.4.1 Вычисление хэш-значения для файлов

Произвести хэширование содержимого файлов и записать результат в "имя_исходного_файла.hsh" можно с помощью команды:

```
-hash [-dir <папка>] [-provtype <N>] [-provname <CSP>]
      [-hashalg <OID>] [-hex] <маска файлов>
```

-dir <папка> папка для файлов с значениями хэш-функции (по умолчанию текущая)

-provtype <N> тип криптопровайдера (по умолчанию 80)

-provname <CSP> имя криптопровайдера

-hashalg <OID> задать алгоритм хэширования при помощи OID, который указывается в поле <OID>:
1.2.643.2.2.9 для ГОСТ Р 34.11-94
1.2.643.7.1.1.2.2 для ГОСТ Р 34.11-2012 256 bit
1.2.643.7.1.1.2.3 для ГОСТ Р 34.11-2012 512 bit

-hex значение хэш-функции в файле в виде шестнадцатеричной строки

<маска файлов> маска для отбора хэшируемых файлов

Примечание. Если опция **-provname** не указана, то будет использован провайдер по умолчанию указанного типа (**-provtype**).
Если указанная в опции **-dir** папка не существует, то она будет создана.

Пример 1. Вычислить для всех файлов с расширением ".txt" из текущей папки значения хэш-функции и записать их в папку "hashes"; использовать провайдер 81-го типа:

```
cryptcp -hash -dir ./hashes -provtype 81 -provname "Crypto-Pro GOST R 34.10-2012 Strong Cryptographic
Service Provider" -hashalg 1.2.643.7.1.1.2.3 *.txt
```

2.4.2 Проверка хэш-значения для файлов

Проверить значение хэша файла, полученное с помощью команды **-hash**, можно с помощью команды:

```
-vhash [-dir <папка>] [-provtype <N>] [-provname <CSP>]
       [-hex] <маска файлов>
```

-dir <папка> папка для файлов с значениями хэш-функции (по умолчанию текущая)

-provtype <N> тип криптопровайдера (по умолчанию 80)

-provname <CSP> имя криптопровайдера

-hex значение хэш-функции в файле в виде шестнадцатеричной строки

<маска файлов> маска для отбора хэшируемых файлов

Примечание. Если опция **-provname** не указана, то будет использован провайдер по умолчанию указанного типа (**-provtype**).

Пример 1. Проверить для всех файлов с расширением ".txt" в текущей папке значения хэш-функции, эталонные значения хранятся в "./hashes"; при хэшировании использовать провайдер 81-го типа:

```
cryptcp -vhash -dir ./hashes -provtype 81 -provname "Crypto-Pro GOST R 34.10-2012 Strong Cryptographic Service Provider" *.txt
```

2.4.3 Создание подписи для файлов

Создать подписи файлов и записать их в файлы "имя_исходного_файла.sgn" можно следующей командой:

```
  -signf [-attached|-detached] [-dir <папка>] <КПС1> <маска файлов> [-cert|-nocert]
        [-addchain] [-crl] [-der] [-strict] [-nostampcert] [-stampchaincheck]
          [-xlongtype1|-cadest|-cadesbes] [-cadestsa <URL>] [-hashalg <OID>]
[-pin <пароль>|-askpin] [-display] [-fext <расширение>] [-keepbadfiles] [-threads <число>]
```

-attached создать присоединённые подписи

-detached создать отсоединённые подписи в отдельных файлах

-dir <папка> папка для создания сообщений (по умолчанию текущая)

<КПС1> КПС автора подписей

<маска файлов> маска для отбора исходных файлов

-cert добавить в сообщения сертификат отправителя

-nocert не добавлять в сообщения сертификат отправителя (по умолчанию)

-addchain добавить полную цепочку сертификата в подпись

-crl добавить в сообщения список отозванных сертификатов

-der использовать формат DER вместо BASE64

-strict использовать однозначное кодирование DER вместо BER

-nostampcert не требовать включения в штамп сертификата службы штампов времени

-stampchaincheck проверить цепочку сертификата в штампе времени

-xlongtype1 создать подписи CAdES-X Long Type 1

-cadest создать подписи CAdES-T

-cadesbes создать подписи CAdES-BES

-cadestsa <URL> служба штампов времени для CAdES-X Long Type 1, CAdES-T, адрес которой указывается в поле <URL> в виде "http://..."

-hashalg <OID> задать алгоритм хэширования при помощи OID, который указывается в поле <OID>:
1.2.643.2.2.9 для ГОСТ Р 34.11-94
1.2.643.7.1.1.2.2 для ГОСТ Р 34.11-2012 256 bit
1.2.643.7.1.1.2.3 для ГОСТ Р 34.11-2012 512 bit

-pin <пароль> задать пароль ключевого контейнера

-askpin запросить пароль ключевого контейнера из консоли

-display вывести на экран информацию от средства доверенного отображения подписываемых данных

-fext <расширение> задать расширение для сообщений; расширения по умолчанию: ".sig" для -attached и ".sgn" для -detached

-keepbadfiles не удалять сообщения при ошибке

-threads <число> число параллельно работающих потоков (по умолчанию 1)

Примечание. Если указанная папка не существует, то она будет создана.

Примечание. Для работы cryptcp с подписью CAdES необходима установка КриптоПро ЭЦП Runtime.

Пример 1. Создать присоединенные подписи CAdES-T всех файлов с расширением ".txt" из текущей папки, используя сертификат с "CN=Test" из хранилища "Личное" ("My") текущего пользователя, добавить в подписи сертификат отправителя; сохранить подписи в папке "sign":

```
cryptcp -signf -attached -dir ./sign -dn "CN=Test" *.txt -cert -cadest -cadestsa http://testca.cryptopro.ru/tsp/tsp.srf
```

Пример 2. Создать отсоединенные подписи всех файлов с расширением ".txt" из текущей папки, используя сертификат с "CN=Test" из хранилища "Личное" ("My") текущего пользователя; сохранить подписи с расширением ".p7s" в папке "sign":

```
cryptcp -signf -detached -dir ./sign -dn "CN=Test" *.txt -fext .p7s
```

2.4.4 Проверка подписи файлов

Проверить подписи содержимого файлов, созданные с помощью предыдущей команды, можно следующим образом:

```
-vsignf -attached [-xlongtype1|-cadest|-cadesbes|-nocades]
        [-threads <число>] <КПС> <маска файлов 1>

    -vsignf [-detached] [-dir <папка>] [-fext <расширение>]
[-xlongtype1|-cadest|-cadesbes|-nocades] [-threads <число>] <КПС> <маска файлов 2>
```

-attached проверить присоединённые подписи

-detached проверить отсоединённые подписи в отдельных файлах

-dir <папка>	папка с сообщениями (по умолчанию текущая)
-xlongtype1	проверить подписи CAdES-X Long Type 1, КПС будет проигнорирован
-cadest	проверить подписи CAdES-T
-cadesbes	проверить подписи CAdES-BES
-nocades	запретить использование вложенных в подписи доказательств
-threads <число>	число параллельно работающих потоков (по умолчанию 1)
<КПС>	КПС автора подписей
<маска файлов 1>	маска файлов с сообщениями
<маска файлов 2>	маска исходных файлов
-fext <расширение>	задать расширение сообщений; расширение по умолчанию: ".sgn"

Примечание. Для работы cryptcp с подписью CAdES необходима установка КриптоПро ЭЦП Runtime.

Пример 1. Проверить присоединенные подписи CAdES-T всех сообщений с расширением ".sig" из папки "sign", используя сертификат с "CN=Test" из хранилища "Личное" ("My") текущего пользователя:

```
cryptcp -vsignf -attached -cadest -dn "CN=Test" ./sign/*.sig
```

Пример 2. Проверить отсоединенные подписи всех файлов с расширением ".txt" из текущей папки, используя сертификат с "CN=Test" из хранилища "Личное" ("My") текущего пользователя; сообщения с расширением ".p7s" находятся в папке "sign":

```
cryptcp -vsignf -detached -dir ./sign -dn "CN=Test" *.txt -fext .p7s
```

2.4.5 Добавление подписи в файлы

Добавить подпись файла в "исходный_файл.sgn" можно командой:

```
cryptcp
    -addsignf -attached <КПС1> <маска файлов 1> [-cert|-nocert] [-crl] [-der]
    [-nostampcert] [-stampchaincheck] [-xlongtype1|-cadest|-cadesbes]
      [-cadestsa <URL>] [-pin <пароль>|-askpin] [-threads <число>]

    -addsignf [-detached] [-dir <папка>] <КПС1> <маска файлов 2> [-cert|-nocert]
[-crl] [-der] [-nostampcert] [-stampchaincheck] [-xlongtype1|-cadest|-cadesbes]
[-cadestsa <URL>] [-pin <пароль>|-askpin] [-threads <число>] [-fext <расширение>]
```

-attached	добавить присоединенные подписи в существующие сообщения
-detached	добавить отсоединенные подписи в существующие сообщения
-dir <папка>	папка с сообщениями; по умолчанию текущая
<КПС1>	КПС автора подписей
<маска файлов 1>	маска файлов с сообщениями
<маска файлов 2>	маска исходных файлов

-cert	добавить в подписи сертификат отправителя
-nocert	не добавлять в сообщения сертификат отправителя (по умолчанию)
-crl	добавить в подписи список отозванных сертификатов
-der	использовать формат DER вместо BASE64
-nostampcert	не требовать включения в штамп сертификата службы штампов времени (используется вместе с -cadest)
-stampchaincheck	проверить цепочку сертификата в штампе времени
-xlongtype1	добавить подписи CAdES-X Long Type 1
-cadest	добавить подписи CAdES-T
-cadesbes	добавить подписи CAdES-BES
-cadestsa <URL>	служба штампов времени для CAdES-X Long Type 1, CAdES-T, адрес которой указывается в поле <URL> в виде "http://..."
-pin <пароль>	задать пароль ключевого контейнера
-askpin	запросить пароль ключевого контейнера из консоли
-threads <число>	число параллельно работающих потоков (по умолчанию 1)
-fext <расширение>	задать расширение сообщений; расширение по умолчанию: ".sgn"

Примечание. Для работы cryptcp с подписью CAdES необходима установка КриптоПро ЭЦП Runtime.

Пример 1. Добавить присоединенные подписи формата CAdES-T для всех сообщений с расширением ".sig" из папки "sign" (использовать сертификат с "CN=Test" из хранилища "Личное" ("My") текущего пользователя); запросить пароль контейнера из консоли:

```
cryptcp -addsignf -attached -dn "CN=Test" ./sign/*.sig -cadest -cadestsa http://testca.cryptopro.ru/
tsp/tsp.srf -askpin
```

Пример 2. Добавить отсоединенные подписи для всех сообщений с расширением ".sig" из папки "sign" (использовать сертификат с "CN=Test" из хранилища "Личное" ("My") текущего пользователя); исходные файлы с расширением ".txt" находятся в текущей папке:

```
cryptcp -addsignf -detached -dir ./sign -dn "CN=Test" *.txt -fext .sig
```

2.4.6 Шифрование файлов

Для того, чтобы создать зашифрованные сообщения, необходимо выполнить следующую команду:

```
-encrf <КПС> [-der] [-strict] [-encryptionalg <OID>] [-keepbadfiles]
 [-nousagecheck] <маска файлов> [-dir <папка>] [-fext <расширение>]
```

<КПС>	КПС получателей
-der	использовать формат DER вместо BASE64

-strict	использовать однозначное кодирование DER вместо BER
-encryptionalg <OID>	задать алгоритм шифрования при помощи OID
-keepbadfiles	не удалять выходной файл при ошибке
-nousagecheck	игнорировать поле сертификата "Использование ключа"
<маска файлов>	маска для отбора файлов для зашифрования
-dir <папка>	папка для создания сообщений (по умолчанию текущая)
-fext <расширение>	задать расширение сообщений; расширение по умолчанию: ".p7e"

Пример 1. Зашифровать содержимое всех файлов с расширением ".txt" из текущей папки, используя алгоритм ГОСТ 28147-89 и сертификат, взятый из файла "cert.p7b", сохранить полученные сообщения с расширением ".enc" в папку "encr":

```
cryptcp -encrf -f cert.p7b -encryptionalg 1.2.643.2.2.21 *.txt -dir ./encr -fext .enc
```

2.4.7 Расшифрование файлов

Для того, чтобы расшифровать сообщения, необходимо выполнить следующую команду:

```
-decrf [<КПС>] [-start] [-pin <пароль>|-askpin] [-keepbadfiles] <маска файлов> [-dir <папка>]
```

<КПС>	КПС получателей
-start	открыть (запустить) полученные файлы
-pin <пароль>	задать пароль ключевого контейнера
-askpin	запросить пароль ключевого контейнера из консоли
-keepbadfiles	не удалять выходной файл при ошибке
<маска файлов>	маска для отбора файлов для расшифрования
-dir <папка>	папка для создания выходных файлов (по умолчанию текущая)

Пример 1. Расшифровать все сообщения с расширением ".p7e" из папки "encr", используя закрытый ключ, связанный с сертификатом из хранилища "Личное" ("My") текущего пользователя, содержащим в поле "Subject" "CN=Ivan Petrov", и сохранить полученные файлы в папку "decr":

```
cryptcp -decrf -dn "CN=Ivan Petrov" ./encr/*.p7e -dir ./decr/
```

2.5 Работа с подписями

2.5.1 Создание подписанного сообщения

Подписать данные и создать сообщение можно следующим образом:

```
-sign <КПС1> [-cert|-nocert] [-addchain] [-crl] [-der] [-strict]
  [-authattr <атрибут>]n раз [-attr <атрибут>]k раз [-nostampcert]
[-stampchaincheck] [-xlongtype1|-cadest|-cadesbes] [-cadestsa <URL>]
  [-hashalg <OID>] [-pin <пароль>|-askpin] [-display|-displayattr]
[-detached|-attached] [-keepbadfiles] <исходный файл> [-fext <расширение>|<сообщение>]
```

<КПС1> КПС автора подписи

-cert добавить в сообщение сертификат отправителя (по умолчанию)

-nocert не добавлять в сообщение сертификат отправителя

-addchain добавить полную цепочку сертификата в подпись

-crl добавить в сообщение список отозванных сертификатов

-der использовать формат DER вместо BASE64

-strict использовать однозначное кодирование DER вместо BER

-authattr <атрибут> добавить подписанный атрибут в подпись; поле <атрибут> заполняется следующим образом: "<OID>,<файл с закодированным содержимым атрибута>" (пример: "1.2.3,attr.bin")

-attr <атрибут> добавить неподписанный атрибут в подпись; поле <атрибут> заполняется следующим образом: "<OID>,<файл с закодированным содержимым атрибута>" (пример: "1.2.3,attr.bin")

-nostampcert не требовать включения в штамп сертификата службы штампов времени (используется вместе с -cadest)

-stampchaincheck проверить цепочку сертификата в штампе времени

-xlongtype1 создать подпись CAdES-X Long Type 1

-cadest создать подпись CAdES-T

-cadesbes создать подпись CAdES-BES

-cadestsa <URL> служба штампов времени для CAdES-X Long Type 1, CAdES-T, адрес которой указывается в поле <URL> в виде "http://..."

-hashalg <OID> задать алгоритм хэширования при помощи OID, который указывается в поле <OID>:
1.2.643.2.2.9 для ГОСТ Р 34.11-94
1.2.643.7.1.1.2.2 для ГОСТ Р 34.11-2012 256 bit
1.2.643.7.1.1.2.3 для ГОСТ Р 34.11-2012 512 bit

-pin <пароль> задать пароль ключевого контейнера

-askpin запросить пароль ключевого контейнера из консоли

-display вывести на экран информацию от средства доверенного отображения подписываемых данных

-displayattr вывести атрибут на экран средства доверенного отображения подписываемых данных

-detached создать отсоединённую подпись в отдельном файле

-attached создать присоединённую подпись

-keepbadfiles не удалять файл с созданным сообщением при ошибке

<исходный файл> файл, содержащий исходные данные

-fext <расширение> задать расширение для сообщения в папке с исходным файлом; расширения по умолчанию: ".sig" для -attached и ".sgn" для -detached

<сообщение> файл, содержащий сообщение

Примечание. Для работы cryptcp с подписью CAdES необходима установка КриптоПро ЭЦП Runtime.

Пример 1. Создать отсоединенную подпись с расширением ".p7s" для файла "in.txt", используя сертификат из хранилища "Личное" ("My") локальной машины, содержащий в поле "Subject" строку "TestUser"; добавить в подпись список отозванных сертификатов и неподписанный атрибут:

```
cryptcp -sign -mmy -dn TestUser -crl -attr "1.2.3,attr.bin" -detached -fext .p7s in.txt
```

Примечание. Поиск используемого сертификата происходит следующим образом:

1. Находятся все сертификаты хранилища "Личные" текущего пользователя и локального компьютера.
2. Если обнаружено более пяти сертификатов, то появляется сообщение об ошибке, иначе пользователю будет предложено выбрать один из найденных сертификатов.

2.5.2 Добавление подписи в сообщение

Добавить электронную подпись в сообщение можно с помощью вызова:

```
    -addsign [-attached] <КПС1> [-cert|-nocert] [-crl] [-nostampcert]
[-stampchaincheck] [-xlongtype1|-cadest|-cadesbes] [-cadestsa <URL>]
[-hashalg <OID>][-pin <пароль>|-askpin] [-authattr <атрибут>]n раз
                [-attr <атрибут>]k раз <сообщение>

   -addsign -detached <КПС1> [-cert|-nocert] [-crl] [-nostampcert]
[-stampchaincheck] [-xlongtype1|-cadest|-cadesbes] [-cadestsa <URL>]
[-hashalg <OID>] [-pin <пароль>|-askpin] [-authattr <атрибут>]n раз
[-attr <атрибут>]k раз <исходный файл> [-fext <расширение>|<сообщение>]
```

-attached добавить присоединенную подпись в существующее сообщение

-detached добавить отсоединенную подпись в существующее сообщение

<КПС1> КПС автора подписи

-cert добавить в сообщение сертификат отправителя (по умолчанию)

-nocert не добавлять в сообщение сертификат отправителя

-crl добавить в сообщение список отозванных сертификатов

-nostampcert не требовать включения в штамп сертификата службы штампов времени (используется вместе с -cadest)

-stampchaincheck проверить цепочку сертификата в штампе времени

-xlongtype1 добавить подпись CAdES-X Long Type 1

-cadest	добавить подпись CAdES-T
-cadesbes	добавить подпись CAdES-BES
-cadestsa <URL>	служба штампов времени для CAdES-X Long Type 1, CAdES-T, адрес которой указывается в поле <URL> в виде "http://..."
-hashalg <OID>	задать алгоритм хэширования при помощи OID, который указывается в поле <OID>: 1.2.643.2.2.9 для ГОСТ Р 34.11-94 1.2.643.7.1.1.2.2 для ГОСТ Р 34.11-2012 256 bit 1.2.643.7.1.1.2.3 для ГОСТ Р 34.11-2012 512 bit
-pin <пароль>	задать пароль ключевого контейнера
-askpin	запросить пароль ключевого контейнера из консоли
-authattr <атрибут>	добавить подписанный атрибут в подпись; поле <атрибут> заполняется следующим образом: "<OID>,<файл с закодированным содержимым атрибута>" (пример: "1.2.3,attr.bin")
-attr <атрибут>	добавить неподписанный атрибут в подпись; поле <атрибут> заполняется следующим образом: "<OID>,<файл с закодированным содержимым атрибута>" (пример: "1.2.3,attr.bin")
<сообщение>	файл, содержащий сообщение
<исходный файл>	файл, содержащий исходные данные
-fext <расширение>	задать расширение для сообщения в папке с исходным файлом; расширение по умолчанию: ".sgn"

Примечание. Для работы cryptcp с подписью CAdES необходима установка КриптоПро ЭЦП Runtime.

Пример 1. Добавить присоединенную подпись CAdES-X Long Type 1 в существующее сообщение "message.sgn", используя сертификат с указанным отпечатком из хранилища "Личное" ("My") текущего пользователя:

```
cryptcp -addsign -attached -thumbprint 9e1fc7cc197abcbfc3b4bf4211d1123a42d78d0f -xlongtype1
-cadestsa http://testca.cryptopro.ru/tsp/tsp.srf message.sgn
```

Пример 2. Добавить отсоединенную подпись в существующее сообщение с расширением ".p7s", которое будет найдено в папке с исходным файлом "in.txt"; для подписи используется сертификат с "CN=Test" из хранилища "Личное" ("My") локальной машины:

```
cryptcp -addsign -detached -mmy -dn "CN=Test" -fext .p7s in.txt
```

2.5.3 Удаление подписи из сообщения

Удалить электронную подпись из сообщения можно командой:

```
-delsign [-attached] <КПС1> <сообщение>

-delsign -detached <КПС1> <исходный файл> [-fext <расширение>|<сообщение>]
```

-attached	удалить присоединённую подпись
-detached	удалить отсоединённую подпись из отдельного файла
<КПС1>	КПС автора подписи
<сообщение>	файл, содержащий сообщение
<исходный файл>	файл, содержащий исходные данные
-fext <расширение>	задать расширение для сообщения в папке с исходным файлом; расширение по умолчанию: ".sgn"

Пример 1. Удалить присоединенную подпись (сертификат автора подписи находится в хранилище "Личное" ("My") и содержит "CN=Test" в поле "Subject") из сообщения "message.sgn":

```
cryptcp -delsign -dn "CN=Test" message.sgn
```

Пример 2. Удалить отсоединенную подпись из сообщения (сертификат автора подписи находится в хранилище "Личное" ("My") и обладает указанным отпечатком), сообщение с расширением ".p7s" будет найдено автоматически в папке с исходным файлом "in.txt":

```
cryptcp -delsign -detached -thumbprint 9e1fc7cc197abcbfc3b4bf4211d1123a42d78d0f in.txt -fext .p7s
```

2.5.4 Проверка подписи

Для проверки электронной подписи в сообщении необходимо воспользоваться командой:

```
-verify [-attached] [<КПС>|-verall] [-start]
  [-xlongtype1|-cadest|-cadesbes|-nocades]
[-keepbadfiles] <сообщение> [<выходной файл>]

-verify -detached [<КПС>|-verall] [-start]
[-xlongtype1|-cadest|-cadesbes|-nocades] <исходный файл>
          [-fext <расширение>|<сообщение>]
```

-attached	проверить присоединённую подпись
-detached	проверить отсоединённую подпись из отдельного файла
<КПС>	КПС авторов подписей
-verall	проверять все подписи (иначе – только подписи авторов из КПС)
-start	открыть (запустить) полученный файл
-xlongtype1	проверить подпись CAdES-X Long Type 1, КПС будет проигнорирован
-cadest	проверить подпись CAdES-T
-cadesbes	проверить подпись CAdES-BES
-nocades	запретить использование вложенных в подпись доказательств
-keepbadfiles	не удалять выходной файл при ошибке
<сообщение>	файл, содержащий сообщение
<выходной файл>	файл, в который будут записаны данные из сообщения

<исходный файл> файл, содержащий исходные данные

-fext <расширение> задать расширение для сообщения в папке с исходным файлом; расширение по умолчанию: ".sgn"

Примечание. Если в сообщении содержится сертификат кого-то из авторов подписей, то используется именно этот сертификат.

Примечание. Для работы cryptcp с подписью CAdES необходима установка КриптоПро ЭЦП Runtime.

Пример 1. Проверить присоединенную подпись "message.sgn", используя сертификат с "CN=Test" из хранилища "Другие пользователи" ("Addressbook") текущего пользователя; сохранить данные из сообщения в файл "output.txt":

```
cryptcp -verify -attached -uaddressbook -dn "CN=Test" message.sgn output.txt
```

Пример 2. Проверить отсоединенную подпись для файла "inputfile.txt", сообщение с расширением ".sgn" будет найдено в папке с исходным файлом "inputfile.txt"; используется сертификат с указанным отпечатком из хранилища "Личное" ("My") текущего пользователя:

```
cryptcp -verify -detached -thumbprint 9e1fc7cc197abcbfc3b4bf4211d1123a42d78d0f inputfile.txt
```

2.5.5 Добавление неподписанного атрибута

Добавить неподписанный атрибут в подпись можно с помощью команды:

```
-addattr <КПС1> [-attr <атрибут>]n раз <сообщение>
```

<КПС1> КПС автора подписи

-attr <атрибут> добавить неподписанный атрибут в подпись; поле <атрибут> заполняется следующим образом: "<OID>,<файл с закодированным содержимым атрибута>" (пример: "1.2.3,attr.bin")

<сообщение> файл, содержащий сообщение

Пример 1. Добавить неподписанный атрибут с указанным OID в сообщение "message.sgn" (сертификат автора подписи содержит "CN=Test" в поле "Subject"):

```
cryptcp -addattr -dn "CN=Test" -attr "1.2.840.113549.1.9.5,attr.bin" message.sgn
```

Примечание. Используется исключительно для добавления неподписанного атрибута в подписанные сообщения. Для текстовых или других файлов не работает.

2.6 Работа с сертификатами

2.6.1 Копирование сертификата в хранилище

Скопировать сертификаты в заданное хранилище можно с помощью команды:

```
-copycert <КПС> [-dm[<название>]|-du[<название>]|-df <файл> [-der]]
```

<КПС> КПС, которые надо скопировать

-dm <название> скопировать в хранилище <название> компьютера (LOCAL_MACHINE); название конечного хранилища по умолчанию: "My"

-du <название> скопировать в хранилище <название> пользователя (CURRENT_USER); название конечного хранилища по умолчанию: "My"

-df <файл> использовать в качестве хранилища файл сертификата

-der использовать формат DER вместо BASE64 (только с ключом -df)

Примечание. Если указан ключ -df, то, в случае, если найден только один сертификат, создается файл типа ".cer", иначе – ".p7b".

Пример 1. Скопировать все сертификаты из хранилища "Личное" ("My") текущего пользователя в файл "mycerts.p7b" (в кодировке BASE64):

```
cryptcp -copycert -u -df mycerts.p7b
```

2.6.2 Копирование сертификата из ключевого контейнера в хранилище

Скопировать сертификат из ключевого контейнера в заданное хранилище можно с помощью следующей команды:

```
-cspcert [-provtype <N>] [-provname <CSP>] [-cont <имя>]
[-ku|-km] [-ex|-sg] [-dm[<название>]|-du[<название>]|-df <файл> [-der]]
```

-provtype <N> тип криптопровайдера (по умолчанию 80)

-provname <CSP> имя криптопровайдера

-cont <имя> имя ключевого контейнера (по умолчанию выбор из списка)

-ku использовать контейнер пользователя (CURRENT_USER)

-km использовать контейнер компьютера (LOCAL_MACHINE)

-ex использовать ключ для обмена зашифрованными данными

-sg использовать ключ для работы с подписями

-dm скопировать в хранилище компьютера (LOCAL_MACHINE)

-dm <название> скопировать в хранилище <название> компьютера (LOCAL_MACHINE); название конечного хранилища по умолчанию: "My"

-du <название> скопировать в хранилище <название> пользователя (CURRENT_USER); название конечного хранилища по умолчанию: "My"

-df <файл> использовать в качестве хранилища файл сертификата

-der использовать формат DER вместо BASE64

Примечание. Если опция **-provname** не указана, то будет использован провайдер по умолчанию указанного типа (**-provtype**). Для операционных систем семейства UNIX в качестве параметра опции **-cont** необходимо указывать имя контейнера вместе со считывателем в формате "\\.\имя_считывателя\имя_контейнера" (например, "\\.\HDIMAGE\cont_name").

Пример 1. Скопировать сертификат из контейнера с именем "cont" (контейнер компьютера) в файл "webserver.cer" (в кодировке ".der"):

```
cryptcp -cspcert -km -cont "\\.\HDIMAGE\cont" -df webserver.cer -der
```

2.6.3 Удаление сертификата из хранилища

Удалить сертификат из хранилища можно командой:

```
-delcert <КПС> [-yes]
```

<КПС> КПС удаляемых сертификатов

-yes автоматически отвечать на все вопросы "Да"

Пример 1. Удалить сертификат, содержащий в поле "Subject" строку "OldServer", из хранилища локальной машины:

```
cryptcp -delcert -m -dn OldServer
```

2.7 Работа с запросами на сертификат

Взаимодействие с удостоверяющим центром должно осуществляться с учетом Регламента УЦ.

Примечание. Ряд функций по взаимодействию с УЦ в зависимости от исполнения и конфигурации средств УЦ могут быть недоступными.

2.7.1 Создание и сохранение запроса сертификата

Команда для создания и сохранения запроса в формате PKCS#10:

```
-createrqst -rdn <RDN> [-provtype <N>] [-provname <CSP>] [-smime]
[-nokeygen|-exprt] [-keysize <n>] [-hashalg <OID>] [-ex|-sg|-both] [-ku|-km]
[-cont <имя>] [-silent] [-pin <пароль>|-askpin] [-certusage <OID>]
[-requestlic] [-der] [-altname <имя>]n раз [-ext <расширение>]n раз <имя файла>
```

-rdn <RDN> список имен полей RDN (например: CN, O, E, L) и их значений: <ИмяПоля1>=<ЗначениеПоля1>[,<ИмяПоля2>=<ЗначениеПоля2>...]

-provtype <N> тип криптопровайдера (по умолчанию 80)

-provname <CSP> имя криптопровайдера

-smime включить возможности S/MIME (только Windows)

-nokeygen использовать существующие ключи из указанного контейнера -cont (если контейнер не указан, выбор из списка)

-exprt	пометить ключи как экспортируемые
-keysize <n>	установить длину ключа (n)
-hashalg <OID>	задать алгоритм хэширования при помощи OID, который указывается в поле <OID>: 1.2.643.2.2.9 для ГОСТ Р 34.11-94 1.2.643.7.1.1.2.2 для ГОСТ Р 34.11-2012 256 bit 1.2.643.7.1.1.2.3 для ГОСТ Р 34.11-2012 512 bit
-ex	создать/использовать ключ для обмена зашифрованными данными; не рекомендуется для сертификатов TLS с назначением 1.3.6.1.5.5.7.3.1 (сервер) или 1.3.6.1.5.5.7.3.2 (клиент)
-sg	создать/использовать ключ для подписи
-both	создать/использовать ключ для обмена с возможностью подписи
-ku	создать/использовать контейнер пользователя (CURRENT_USER)
-km	создать/использовать контейнер компьютера (LOCAL_MACHINE)
-cont <имя>	задать имя ключевого контейнера
-silent	не выводить графические окна на экран при выполнении команды
-pin <пароль>	пароль ключевого контейнера
-askpin	запросить пароль ключевого контейнера из консоли (только UNIX)
-certusage <OID>	задать назначения сертификата (OID) через запятую (например, "1.3.6.1.5.5.7.3.4,1.3.6.1.5.5.7.3.2")
-requestlic	запросить сертификат со встроенной лицензией на КриптоПро CSP; УЦ должен быть настроен на выдачу таких сертификатов
-der	использовать формат DER вместо BASE64
-altname <имя>	добавить альтернативное имя субъекта (DNS-name) к запросу
-ext <расширение>	добавить расширение к запросу; в поле <расширение> указывается имя файла с закодированным расширением (BASE64 или DER)
<имя файла>	имя файла, в котором следует сохранить запрос

Примечание. Если опция **-provname** не указана, то будет использован провайдер по умолчанию указанного типа (**-provtype**). Далее, если не указаны опции **-nokeygen** и **-cont**, то имя контейнера сгенерирует криптопровайдер. Для операционных систем семейства UNIX в качестве параметра опции **-cont** необходимо указывать имя контейнера вместе со считывателем в формате "\\.\имя_считывателя\имя_контейнера" (например, "\\.\HDIMAGE\cont_name").

Пример 1. Создать запрос на сертификат с "E=ivanov@bank.ru,CN=Иванов" и сгенерировать контейнер с именем cont на указанном носителе, используя криптопровайдер "Crypto-Pro GOST R 34.10-2012 Strong Cryptographic Service Provider", сохранить запрос в файл "myrequest.req":

```
cryptcp -createrqst -rdn "E=ivanov@bank.ru,CN=Иванов" -provtype 81 -provname "Crypto-Pro GOST R
34.10-2012 Strong Cryptographic Service Provider" -cont "\\.\HDIMAGE\cont" myrequest.req
```

Пример 2. Создать запрос на сертификат с "E=ivanov@bank.ru,CN=Иванов", задав пароль ключевого контейнера

"123" и альтернативные имена субъекта "87.250.251.12" и "example.com", используя криптопровайдер "Crypto-Pro GOST R 34.10-2012 Cryptographic Service Provider" и алгоритм хэширования ГОСТ Р 34.11-2012 256 bit, и сохранить его в файл "myrequest.req":

```
cryptcp -createrqst -rdn "E=ivanov@bank.ru,CN=Иванов" -provname "Crypto-Pro GOST R 34.10-2012
Cryptographic Service Provider" -hashalg 1.2.643.7.1.1.2.2 -pin 123 -altname 87.250.251.12
-altname example.com myrequest.req
```

2.7.2 Установка сертификата из файла

Установка сертификата из файла PKCS#7 или файла сертификата осуществляется с помощью команды:

```
-instcert [-provtype <N>] [-provname <CSP>] [-cont <имя>] [-ku|-km]
[-dm[<название>]|du[<название>]][-nocsp|-from_request] [-pin <пароль>|-askpin]
[-enable-install-root] <имя файла>
```

-provtype <N> тип криптопровайдера (по умолчанию 80)

-provname <CSP> имя криптопровайдера

-cont <имя> задать имя ключевого контейнера (по умолчанию выбор из списка)

-ku использовать контейнер пользователя (CURRENT_USER)

-km использовать контейнер компьютера (LOCAL_MACHINE)

-dm <название> установить в хранилище <название> компьютера (LOCAL_MACHINE); название конечного хранилища по умолчанию: "My"

-du <название> установить в хранилище <название> пользователя (CURRENT_USER); название конечного хранилища по умолчанию: "My"

-nocsp не устанавливать сертификат в контейнер

-from_request выбрать контейнер в соответствии со сделанным ранее запросом

-pin <пароль> пароль ключевого контейнера

-askpin запросить пароль ключевого контейнера из консоли (только UNIX)

<имя файла> имя файла, содержащего сертификат

-enable-install-root не запрашивать разрешение на установку корневого сертификата в хранилище "Доверенные корневые центры" (только на UNIX с -dm)

Примечание. Если указана опция **-nocsp**, то опции **-provname, -provtype, -cont, -km, -ku** игнорируются. Если опция **-provname** не указана, то будет использован провайдер по умолчанию указанного типа (**-provtype**). Для операционных систем семейства UNIX в качестве параметра опции **-cont** необходимо указывать имя контейнера вместе со считывателем в формате "\\.\имя_считывателя\имя_контейнера" (например, "\\.\HDIMAGE\cont_name").

Пример 1. Установить сертификат "mycert.cer" в контейнер "cont" на указанном носителе, а также в хранилище "Личное" ("My") текущего пользователя с привязкой к контейнеру:

```
cryptcp -instcert -cont "\\.\HDIMAGE\cont" mycert.cer
```

2.7.3 Просмотр настроек учетных записей пользователей УЦ

Получить информацию о настройках параметров учетных записей пользователя на УЦ можно с помощью команды:

```
-listdn [-cpca <адрес УЦ>|-cpca20 <адрес УЦ>]
```

-cpca <адрес УЦ> указать адрес веб-интерфейса КриптоПро УЦ вида "http://xxx.yyy/zzz" или "https://xxx.yyy/zzz", иначе это адрес "CP CSP Test CA": https://cryptopro.ru/ui

-cpca20 <адрес УЦ> указать адрес веб-интерфейса КриптоПро УЦ вида "https://xxx.yyy/zzz/<folder>" (folder обозначает GUID папки УЦ или путь папки в иерархии папок, при этом разделителем имен папок в пути является символ '|'), иначе это адрес "CP CSP Test CA": https://cryptopro.ru/ui

Пример 1. Получить информацию о настройках параметров учетных записей пользователей на УЦ:

```
cryptcp -listdn -cpca20 https://testca2012.cryptopro.ru/ui/1e09b6ed-01c3-481a-bdc4-a9ea00996da4
```

2.7.4 Регистрация пользователя на УЦ

Регистрация пользователя на УЦ осуществляется с помощью команды:

```
-createuser [-cpca <адрес УЦ>|-cpca20 <адрес УЦ>] [-field <ID поля>]n раз
```

-cpca <адрес УЦ> указать адрес веб-интерфейса КриптоПро УЦ вида "http://xxx.yyy/zzz" или "https://xxx.yyy/zzz", иначе это адрес "CP CSP Test CA": https://cryptopro.ru/ui

-cpca20 <адрес УЦ> указать адрес веб-интерфейса КриптоПро УЦ вида "https://xxx.yyy/zzz/<folder>" (folder обозначает GUID папки УЦ или путь папки в иерархии папок, при этом разделителем имен папок в пути является символ '|'), иначе это адрес "CP CSP Test CA": https://cryptopro.ru/ui

-field <ID поля> добавить поле в DN регистрируемого пользователя; список идентификаторов поля DN можно посмотреть командой `cryptcp -listdn` (см. разд. 2.7.3)

Примечание. При успешном выполнении команда возвращает маркер временного доступа для аутентификации на УЦ КриптоПро и пароль к маркеру временного доступа.

Пример 1. Зарегистрировать пользователя с указанным RDN на УЦ:

```
cryptcp -createuser -field "CN=Test" -field "O=CRYPTO-PRO" -cpca20 https://testca2012.cryptopro.ru/
ui/1e09b6ed-01c3-481a-bdc4-a9ea00996da4
```

2.7.5 Проверка регистрации пользователя на УЦ

Проверить, зарегистрирован ли пользователь на УЦ, можно с помощью команды:

```
-checkreg -token <ID маркера> -tpassword <пароль>
      [-cpca <адрес УЦ>|-cpca20 <адрес УЦ>]
```

-token <ID маркера> задать маркер временного доступа для проверки статуса; <ID маркера> - логин учетной записи на КриптоПро УЦ

-tpassword <пароль> задать пароль к маркеру временного доступа

-cpca <адрес УЦ> указать адрес веб-интерфейса КриптоПро УЦ вида "http://xxx.yyy/zzz" или "https://xxx.yyy/zzz", иначе это адрес "CP CSP Test CA": https://cryptopro.ru/ui

-cpca20 <адрес УЦ> указать адрес веб-интерфейса КриптоПро УЦ вида "https://xxx.yyy/zzz/", иначе это адрес "CP CSP Test CA": https://cryptopro.ru/ui

Пример 1. Проверить регистрацию пользователя testuser на УЦ КриптоПро 2.0:

```
cryptcp -checkreg -token testuser -tpassword 1352443260 -cpca20 https://testca2012.cryptopro.ru/ui
```

2.7.6 Просмотр списка шаблонов сертификатов, доступных пользователю УЦ

```
-listtmpl -cpca20 <адрес УЦ>
[-token <ID маркера> -tpassword <пароль>|-clientcert <КПС1>]
```

-cpca20 <адрес УЦ> указать адрес веб-интерфейса КриптоПро УЦ вида "https://xxx.yyy/zzz/"

-token <ID маркера> использовать маркер временного доступа для аутентификации; <ID маркера> - логин учетной записи на КриптоПро УЦ

-tpassword <пароль> задать пароль к маркеру временного доступа

-clientcert <КПС1> использовать сертификат для аутентификации на КриптоПро УЦ; <КПС1> - КПС пользователя (используется хранилище "Личное" ("My"), по умолчанию текущего пользователя)

Пример 1. Получить список шаблонов, доступных пользователю УЦ; авторизоваться на УЦ по сертификату с "CN=Test" из хранилища "Личное" ("My") текущего пользователя:

```
cryptcp -listtmpl -cpca20 https://testca2012.cryptopro.ru/ui/ -clientcert -dn "CN=Test"
```

2.7.7 Создание запроса, получение и установка сертификата

Создать запрос на сертификат, отправить его в центр сертификации, получить выписанный сертификат и установить его можно с помощью команды:

```
-createcert -rdn <RDN> [-provtype <N>] [-provname <CSP>] [-cont <имя>]
[-nokeygen|-exprt] [-keysize <n>] [-ex|-sg|-both] [-ku|-km] [-hashalg <OID>]
[-certusage <OID>] [-ca <адрес УЦ>|-cpca <адрес УЦ>|-cpca20 <адрес УЦ>] [-smime]
[-requestlic] [-token <ID маркера> -tpassword <пароль>|-clientcert <КПС1>]
[-tmpl <шаблон>] [-dm[<название>]|du[<название>]] [-nocsp] [-silent]
[-pin <пароль>|-askpin] [-fileid <имя файла>] [-altname <имя>]n раз
[-ext <расширение>]n раз [-enable-install-root] [-file <имя>]
```

-rdn <RDN> список имен полей RDN (например: CN, O, E, L) и их значений: <ИмяПоля1>=<ЗначениеПоля1>[,<ИмяПоля2>=<ЗначениеПоля2>...]

-provtype <N> тип криптопровайдера (по умолчанию 80)

-provname <CSP> имя криптопровайдера

-cont <имя> задать имя ключевого контейнера (по умолчанию выбор из списка)

-nokeygen использовать существующие ключи из указанного контейнера -cont (если контейнер не указан, выбор из списка)

-exprt пометить ключи как экспортируемые

-keysize <n> указать длину ключа (n)

-ex создать/использовать ключ для обмена зашифрованными данными; не рекомендуется для сертификатов TLS с назначением 1.3.6.1.5.5.7.3.1 (сервер) или 1.3.6.1.5.5.7.3.2 (клиент)

-sg создать/использовать ключ для подписи

-both создать/использовать ключ для обмена с возможностью подписи

-ku создать/использовать контейнер пользователя (CURRENT_USER)

-km создать/использовать контейнер компьютера (LOCAL_MACHINE)

-hashalg <OID> задать алгоритм хэширования при помощи OID, который указывается в поле <OID>:
1.2.643.2.2.9 для ГОСТ Р 34.11-94
1.2.643.7.1.1.2.2 для ГОСТ Р 34.11-2012 256 bit
1.2.643.7.1.1.2.3 для ГОСТ Р 34.11-2012 512 bit

-certusage <OID> задать назначения сертификата (OID) через запятую (например, "1.3.6.1.5.5.7.3.4,1.3.6.1.5.5.7.3.2")

-ca <адрес УЦ> указать адрес УЦ Microsoft вида "http://xxx.yyy/zzz" или "\\сервер\имяУЦ\" (см. разд. 1), иначе это адрес "CP CSP Test CA": http://www.cryptopro.ru/certsrv/

-cpca <адрес УЦ> указать адрес веб-интерфейса КриптоПро УЦ вида "http://xxx.yyy/zzz" или "https://xxx.yyy/zzz"

-cpca20 <адрес УЦ> указать адрес веб-интерфейса КриптоПро УЦ вида "https://xxx.yyy/zzz/"

-smime включить возможности S/MIME (только Windows)

-requestlic запросить сертификат со встроенной лицензией на КриптоПро CSP; УЦ должен быть настроен на выдачу таких сертификатов

-token <ID маркера> использовать маркер временного доступа для аутентификации; <ID маркера> - логин учетной записи на КриптоПро УЦ

-tpassword <пароль> задать пароль к маркеру временного доступа

-clientcert <КПС1> использовать сертификат для аутентификации на КриптоПро УЦ; <КПС1> - КПС пользователя (используется хранилище "Личное" ("My"), по умолчанию текущего пользователя)

-tmpl <шаблон> задать шаблон запрашиваемого сертификата; <шаблон> - название или OID шаблона (только для УЦ КриптоПро версии 2.0)

-dm <название> установить в хранилище <название> компьютера (LOCAL_MACHINE); название конечного хранилища по умолчанию: "My"

-du <название> установить в хранилище <название> пользователя (CURRENT_USER); название конечного хранилища по умолчанию: "My"

-nocsp	не устанавливать сертификат в контейнер
-silent	не выводить графические окна на экран при выполнении команды
-pin <пароль>	пароль ключевого контейнера
-askpin	запросить пароль ключевого контейнера из консоли (только UNIX)
-fileid <имя файла>	указать имя файла для записи идентификатора запроса в случае "отложенной выдачи" сертификата (см. разд. 2.7.8); если -fileid не указан, то идентификатор будет выведен на экран
-altname <имя>	добавить альтернативное имя субъекта (DNS-name) к запросу
-ext <расширение>	добавить расширение к запросу; в поле <расширешие> указывается имя файла с закодированным расширением (BASE64 или DER)
-enable-install-root	не запрашивать разрешение на установку корневого сертификата в хранилище "Доверенные корневые центры" (только на UNIX с -dm)
-file <имя>	сохранить ответ УЦ в файл (".cer"/".p7b")

Примечание. Если опция **-provname** не указана, то будет использован провайдер по умолчанию указанного типа (**-provtype**). Если не указаны опции **-nokeygen** и **-cont**, то имя контейнера сгенерирует криптопровайдер. Для операционных систем семейства UNIX в качестве параметра опции **-cont** необходимо указывать имя контейнера вместе со считывателем в формате "\\.\имя_считывателя\имя_контейнера" (например, "\\.\HDIMAGE\cont_name").

Пример 1. Выпустить сертификат по шаблону Пользователь с "E=ivanov@bank.ru,CN=Иванов Петр" на УЦ КриптоПро 2.0 (авторизоваться по токену временного доступа), сгенерировать контейнер с именем "cont" на указанном носителе, используя криптопровайдер "Crypto-Pro GOST R 34.10-2012 Strong Cryptographic Service Provider":

```
cryptcp -createcert -rdn "E=ivanov@bank.ru,CN=Иванов" -provtype 81 -provname "Crypto-Pro GOST R
34.10-2012 Strong Cryptographic Service Provider" -cont "\\.\HDIMAGE\cont" -cpca20
https://testca2012.cryptopro.ru/ui -token testuser -tpassword 1352443260 -tmpl User
```

Пример 2. Выпустить сертификат с "E=ivanov@bank.ru,CN=Иванов Петр" на УЦ КриптоПро 2.0 (авторизоваться по токену временного доступа), используя существующий ключ из одного из пользовательских контейнеров, используя криптопровайдер "Crypto-Pro GOST R 34.10-2012 Cryptographic Service Provider". При этом задать назначение сертификата "1.3.6.1.5.5.7.3.4", альтернативные имена субъекта "87.250.251.12" и "example.com" и сохранить ответ УЦ в файл "response.cer":

```
cryptcp -createcert -rdn "E=ivanov@bank.ru,CN=Иванов" -provname "Crypto-Pro GOST R 34.10-2012
Cryptographic Service Provider" -nokeygen -ku -certusage 1.3.6.1.5.5.7.3.4 -cpca20
https://testca2012.cryptopro.ru/ui -token testuser -tpassword 1352443260 -altname 87.250.251.12
-altname example.com -file response.cer
```

2.7.8 Проверка выпуска сертификата, получение и установка сертификата

Проверить, не выпущен ли сертификат, запрос на который был отправлен ранее, получить выписанный сертификат и установить его можно с помощью команды:

```
-pendcert [-provtype <N>] [-provname <CSP>] [-cont <имя>] [-ku|-km]
 [-ca <адрес УЦ>|-cpca <адрес УЦ>|-cpca20 <адрес УЦ>] [-pin <пароль>|-askpin]
[-dm[<название>]|du[<название>]] [-fileid <имя файла>|-requestid <ID запроса>]
    [-token <ID маркера> -tpassword <пароль>|-clientcert <КПС1>] [-nocsp]
                        [-enable-install-root]
```

-provtype <N>	тип криптопровайдера (по умолчанию 80)
-provname <CSP>	имя криптопровайдера
-cont <имя>	задать имя ключевого контейнера (по умолчанию выбор из списка)
-ku	использовать контейнер пользователя (CURRENT_USER)
-km	использовать контейнер компьютера (LOCAL_MACHINE)
-ca <адрес УЦ>	указать адрес УЦ Microsoft вида "http://xxx.yyy/zzz" или "\\сервер\имяУЦ\" (см. разд. 1), иначе это адрес "CP CSP Test CA": http://www.cryptopro.ru/certsrv/
-cpca <адрес УЦ>	указать адрес веб-интерфейса КриптоПро УЦ вида "http://xxx.yyy/zzz" или "https://xxx.yyy/zzz"
-cpca20 <адрес УЦ>	указать адрес веб-интерфейса КриптоПро УЦ вида "https://xxx.yyy/zzz/"
-pin <пароль>	пароль ключевого контейнера
-askpin	запросить пароль ключевого контейнера из консоли (только UNIX)
-dm <название>	установить в хранилище <название> компьютера (LOCAL_MACHINE); название конечного хранилища по умолчанию: "My"
-du <название>	установить в хранилище <название> пользователя (CURRENT_USER); название конечного хранилища по умолчанию: "My"
-fileid <имя файла>	указать имя файла, содержащего идентификатор запроса
-requestid <ID запроса>	указать идентификатор запроса на сертификат
-token <ID маркера>	использовать маркер временного доступа для аутентификации; <ID маркера> - логин учетной записи на КриптоПро УЦ
-tpassword <пароль>	задать пароль к маркеру временного доступа
-clientcert <КПС1>	использовать сертификат для аутентификации на КриптоПро УЦ; <КПС1> - КПС пользователя (используется хранилище "Личное" ("My"), по умолчанию текущего пользователя)
-nocsp	не устанавливать сертификат в контейнер хранилище "Доверенные корневые центры" (только на UNIX с -dm)
-enable-install-root	не запрашивать разрешение на установку корневого сертификата в хранилище "Доверенные корневые центры" (только на UNIX с -dm)

Примечание. Если указана опция **-noCSP**, то опции **-provname, -provtype, -cont, -km, -ku** игнорируются. Если опция **-provname** не указана, то будет использован провайдер по умолчанию указанного типа (**-provtype**). Для операционных систем семейства UNIX в качестве параметра опции **-cont** необходимо указывать имя контейнера вместе со считывателем в формате "\\.\имя_считывателя\имя_контейнера" (например, "\\.\HDIMAGE\cont_name").

Пример 1. Проверить выпуск сертификата с "E=ivanov@bank.ru,CN=Иванов Петр", запрос на который был отправлен ранее на УЦ КриптоПро 2.0 (авторизоваться по токену временного доступа), установить сертификат в контейнер с именем "cont", а также в хранилище "Личное" ("My") текущего пользователя, используя криптопровайдер "Crypto-Pro GOST R 34.10-2012 Cryptographic Service Provider":

```
cryptcp -pendcert -rdn "E=ivanov@bank.ru,CN=Иванов Петр" -cont "\\.\HDIMAGE\cont" -cpca20
https://testca2012.cryptopro.ru/ui -token testuser -tpassword 1352443260 -requestid
853e4fd4-be81-e711-80d5-00155d454d12
```

2.8 Ввод серийного номера лицензии (только для Windows)

Для ввода или отображения серийного номера лицензии используется команда:

```
-sn [<серийный номер>]
```

<серийный номер> серийный номер, который необходимо ввести (можно указывать как с разделителями, так и без них)

Примечание. В операционных системах семейства UNIX команда не используется.

Пример 1. Ввести указанный серийный номер лицензии на компьютере:

```
cryptcp -sn XXXXX-XXXXX-XXXXX-XXXXX-XXXXX
```

2.9 Усовершенствованная электронная подпись

Приложение командной строки поддерживает возможность создания улучшенной электронной подписи, соответствующей стандарту CAdES (см. RFC 5126). Использование формата усовершенствованной подписи имеет значительные преимущества, обеспечивая:

- доказательство момента подписи документа и действительности сертификата ключа подписи на этот момент;
- отсутствие необходимости сетевых обращений при проверке подписи;
- архивное хранение электронных документов;
- простоту встраивания.

Для доказательства момента подписи используются штампы времени, соответствующие международной рекомендации "Internet X.509 Public Key Infrastructure Time-Stamp Protocol (TSP)" (см. RFC 3161), а также методической рекомендации МР 26.2.001-2021 «Информационная технология. Криптографическая защита информации. Использование российских криптографических алгоритмов в протоколе штампов времени (TSP)» Технического комитета по стандартизации «Криптографическая защита информации» (ТК26).

Доказательства действительности сертификата в момент подписи обеспечиваются вложением в реквизиты документа цепочки сертификатов до доверенного УЦ и OCSP-ответов. На эти доказательства также получается штамп времени, подтверждающий их целостность в момент проверки.

При таких условиях появляется возможность проверить подпись в режиме отсутствия сетевых соединений, доступа к службам OCSP и службам штампов времени. Также вся дополнительная информация хранится в реквизитах файла подписи, что требуется для архивного хранения электронных документов.

Для использования формата усовершенствованной подписи реализована возможность применения специальных параметров при создании, добавлении и проверке электронных подписей.

Следующие атрибуты можно использовать при работе с подписями:

-xlongtype1	используется формат подписи CAdES-X Long Type 1
-cadest	используется формат подписи CAdES-T
-cadesbes	используется формат подписи CAdES-BES
-cadesTSA	указывается служба штампов времени для подписи CAdES-X Long Type 1, CAdES-T
-nocades	исключается использование вложенных в подпись доказательств

Примечание. Для работы с усовершенствованной подписью необходимо наличие на компьютере пользователя ПО КриптоПро TSP Client и КриптоПро OCSP Client с действующими лицензиями, которые вводятся через оснастку Управление лицензиями КриптоПро PKI.

Пример 1. Создать подпись формата CAdES-X Long Type 1 для файла "test.txt", используя закрытый ключ, связанный с сертификатом хранилища "Личные" ("My") текущего пользователя, содержащим в поле "Субъект" ("Subject") подстроку "Иванов Петр", с проверкой цепочки найденных сертификатов, используя службу штампов времени http://tsp.test/tsp_root/tsp.srf, и сохранить результат в файл "test.txt.logn_sgn":

```
cryptcp -sign -dn "CN=Иванов Петр" -cadesTSA http://tsp.test/tsp_root/tsp.srf -xlongtype1
C:\data\test.txt C:\data\test.txt.logn_sgn
```

3 Возвращаемые значения

В случае успешного выполнения команды `cryptcp` возвращает 0 (0x00000000). Ненулевое возвращаемое значение обозначает наличие ошибки. Перечень возвращаемых ошибок с соответствующими кодами представлен в табл. 2.

Таблица 2. Коды ошибок cryptcp

Код ошибки (DEC)	**Код ошибки (HEX)**	**Описание ошибки**
536871012	20000064	Мало памяти
536871013	20000065	Не удалось открыть файл
536871014	20000066	Операция отменена пользователем
536871015	20000067	Некорректное преобразование BASE64
536871016	20000068	Если указан параметр '-help', то других быть не должно
536871017	20000069	Файл слишком большой
536871024	20000070	Произошла внутренняя ошибка
536871112	200000C8	Указан лишний файл
536871113	200000C9	Указан неизвестный ключ
536871114	200000CA	Указана лишняя команда
536871115	200000CB	Для ключа не указан параметр
536871116	200000CC	Не указана команда
536871117	200000CD	Не указан необходимый ключ
536871118	200000CE	Указан неверный ключ
536871119	200000CF	Параметром ключа '-q' должно быть натуральное число
536871120	200000D0	Не указан входной файл
536871121	200000D1	Не указан выходной файл
536871122	200000D2	Команда не использует параметр с именем файла
536871123	200000D3	Не указан файл сообщения
536871212	2000012C	Не удалось открыть хранилище сертификатов
536871213	2000012D	Сертификаты не найдены
536871214	2000012E	Найдено более одного сертификата (ключ '-1')
536871215	2000012F	Команда подразумевает использование только одного сертификата
536871216	20000130	Неверно указан номер
536871217	20000131	Нет используемых сертификатов

536871218	20000132	Данный сертификат не может применяться для этой операции
536871219	20000133	Цепочка сертификатов не проверена
536871220	20000134	Криптопровайдер, поддерживающий необходимый алгоритм, не найден
536871221	20000135	Ошибка при вводе пароля на контейнер
536871222	20000136	Не удалось получить закрытый ключ, соответствующий сертификату
536871312	20000190	Не указана маска файлов
536871313	20000191	Указаны несколько масок файлов
536871314	20000192	Файлы не найдены
536871315	20000193	Задана неверная маска
536871316	20000194	Неверный хэш
536871412	200001F4	Ключ '-start' указан, а выходной файл нет
536871413	200001F5	Содержимое файла - не подписанное сообщение
536871414	200001F6	Неизвестный алгоритм подписи
536871415	200001F7	Сертификат автора подписи не найден
536871416	200001F8	Подпись не найдена
536871417	200001F9	Подпись не верна
536871418	200001FA	Штамп времени не верен
536871512	20000258	Содержимое файла — не зашифрованное сообщение
536871513	20000259	Неизвестный алгоритм шифрования
536871514	2000025A	Не найден сертификат с соответствующим закрытым ключом (ключом ЭП)
536871612	200002BC	Не удалось инициализировать COM
536871613	200002BD	Контейнеры не найдены
536871614	200002BE	Не удалось получить ответ от сервера
536871615	200002BF	Сертификат не найден в ответе сервера
536871616	200002C0	Файл не содержит идентификатор запроса
536871617	200002C1	Некорректный адрес ЦС
536871618	200002C2	Получен неверный Cookie
536871619	200002C3	ЦС отклонил запрос
536871620	200002C4	Ошибка при инициализации CURL
536871712	20000320	Серийный номер содержит недопустимое количество символов
536871713	20000321	Неверный код продукта

536871714	20000322	Не удалось проверить серийный номер
536871715	20000323	Не удалось сохранить серийный номер
536871716	20000324	Не удалось загрузить серийный номер
536871717	20000325	Лицензия просрочена

Примечание. Кроме кодов, приведенных в таблице, приложение может возвращать код любой системной ошибки Windows.