127 018, Москва, Сущевский вал, д.18
Телефон: (495) 995 4820
Факс: (495) 995 4820
http://www.CryptoPro.ru
E-mail: info@CryptoPro.ru

КРИПТОПРО

Средство Криптографической Защиты Информации

КриптоПро CSP

Версия 4.0 R4 KC1

Приложение командной строки для подписи и шифрования файлов

ЖТЯИ.00087-03 93 01

Листов 22

2018 г.

Содержание

Аннотация

Данный документ содержит общую информацию по использованию программного продукта «ЖТЯИ.00087-03 93 01. КриптоПро CSP. Приложение командной строки», 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

Windows

Включает программно-аппаратные среды:

Windows XP[1] (x86);
Windows 7/8/8.1/10/Server 2003/2008 (x86, x64);
Windows Server 2008 R2/2012/2012 R2/2016 (x64).

LSB Linux

Включает дистрибутивы Linux, удовлетворяющие стандарту Linux Standard Base ISO/IEC 23360 версии LSB 4.x:

CentOS 4/5/6 (x86, x64);
CentOS 7 (x86, x64, POWER, ARM, ARM64);
ОСь (OS-RT) (x64);
ТД ОС АИС ФССП России (GosLinux) (x86, x64);
Red OS (x86, x64);
Fedora 27/28/29 (x86, x64, ARM);
Oracle Linux 4/5/6 (x86, x64);
Oracle Linux 7 (x64);
OpenSUSE Leap 42, 15 (x86, x64, ARM, ARM64);
AlterOS (x64);
SUSE Linux Enterprise Server 11SP4 (x86, x64);
SUSE Linux Enterprise Server 12/15, Desktop 12/15 (x64, POWER, ARM64);
Red Hat Enterprise Linux 4/5/6 (x86, x64);
Red Hat Enterprise Linux 7 (x64, POWER, ARM64);
Синтез-ОС.РС (x86, x64);
ПК «СинтезМ-Клиент» в составе КП «ЗОС «СинтезМ» (x64);
ПК «СинтезМ-Сервер» в составе КП «ЗОС «СинтезМ» (x64);
КП «ОС «СинтезМ-К» (x64);
Ubuntu 14.04/16.04 (x86, x64, POWER, ARM, ARM64);
Ubuntu 18.04/18.10 (x86, x64);
Linux Mint 17/18/19 (x86, x64);
Debian 7/8/9 (x86, x64, POWER, ARM, ARM64, MIPS);
ОС Лотос (x86, x64);
Astra Linux Special Edition, Common Edition (x64, MIPS, Эльбрус);
МСВСфера 6.3 Сервер (x64, ARM64).

Unix

Включает программно-аппаратные среды:

ОС Эльбрус версия 3 (Эльбрус);
ALT Linux 6/7 (x86, x64, ARM);
Альт Сервер 8, Альт 8 СП Сервер (x86, x64, ARM, ARM64);
Альт Рабочая станция 8, Альт Рабочая станция К 8, Альт 8 СП Рабочая станция (x86, x64, ARM, ARM64);
ROSA Fresh, Enterprise Desktop, Enterprise Linux Server (x86, x64);
РОСА ХРОМ/КОБАЛЬТ/НИКЕЛЬ (x86, x64);

FreeBSD 11, pfSense 2.x (x86, x64);
AIX 6/7 (POWER);
Mac OS X 10.9/10.10/10.11/10.12/10.13/10.14 (x64).

Solaris

Включает программно-аппаратные среды:

Solaris 10 (sparc, x86, x64);
Solaris 11 (sparc, x64).

Sailfish

Включает программно-аппаратную среду:

SailfishOS 2.1.1.12 (ARMv7).

iOS

Включает программно-аппаратные среды:

Apple iOS 8.0/8.0.1/8.0.2/8.1/8.1.1/8.1.2/8.1.3/8.2/8.3/8.4/8.4.1/9/9.0.1/9.0.2/9.1/9.2/9.2.1/9.3/9.3.1/9.3.2/9.3.3/9.3.4/9.3.5/10/11/12 (ARMv7, ARM64).

Виртуальные среды

Microsoft Hyper-V Server 2008/2008R2/2012/2012R2/2016 (x64);
Microsoft Hyper-V 8/8.1/10 (x64);
Citrix XenServer 7 (x64);
VMWare WorkStation 11/12/14/15 (x86, x64);
VMWare WorkStation Player 12/14/15 (x86, x64);
VMWare vSphere ESXi/Hypervisor 5.5/6.0/6.5/6.7 (x64);
Oracle VirtualBox 5.2 (x86, x64);
RHEV 4 (x64).

Примечания:

1. Версия POSReady.

2. Использование программы

2.1. Запуск программы

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

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

путь путь к месторасположению программы (например, «c:\utils\»);

cryptcp имя исполняемого файла приложения;

команда одна из допустимых команд (см. ниже);

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.**

При описании опций звездочкой (*) помечена опция по умолчанию (для нескольких взаимоисключающих опций).

2.2. Критерий поиска сертификатов

Критерий поиска сертификатов (далее – КПС) используется для задания сведений о субъектах, чьи сертификаты будут использоваться при выполнении команды (например, шифрование или подпись 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 <RDN>]m раз[-{m|u}[<имя>]|-f <файл>]k раз [-thumbprint <отпечаток>] [-all|-1|-q[N]] [{-nochain|-errchain [-norev]}]

-dn указание строк для поиска в RDN (иначе поиск не зависит от RDN). Если вводится несколько строк для поиска, то будет найдено большее количество сертификатов;

RDN список строк (через запятую), используемых для поиска сертификатов. будут найдены сертификаты, в RDN субъекта/издателя которых присутствуют все эти строки.

-issuer используется RDN издателя для поиска

-m поиск осуществляется в хранилищах компьютера (LOCAL_MACHINE);

-u* поиск осуществляется в хранилищах пользователя (CURRENT_USER);

имя название хранилища (по умолчанию «My» для создания подписи или расшифровки и «My+Addressbook» для остальных случаев);

-f в качестве хранилища используется сообщение или файл сертификата;

файл имя файла;

-thumbprint отпечаток сертификата;

-all*	использовать все найденные сертификаты (* для КПС);
-1*	будет найден только один сертификат, иначе – ошибка (* для КПС1);
-q[N]	если найдено менее N сертификатов, то вывести запрос для выбора нужного (по умолчанию N=10);
-nochain	не проверять цепочки найденных сертификатов;
-norev	не проверять сертификаты в цепочке на предмет отозванности;
-errchain	завершать выполнение с ошибкой, если хотя бы один сертификат не прошел проверку. 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.

П р и м е р :
Иван Иванов,a@b.c – неверно;
“Иван Иванов,a@b.c” – верно;
CN=Иванов,E=a@b.c – верно.

2.3. Команды шифрования/расшифрования

Для того, чтобы зашифровать данные и создать сообщение, необходимо выполнить следующую команду:

-encr <КПС> [-der] [- strict] [-encryptionAlg <OID>] <входной файл> <сообщение>

КПС	КПС получателей;
-der	использовать формат DER вместо BASE64;
-strict	Использовать однозначное кодирование DER (а не BER);
-encryptionAlg	задать алгоритм шифрования
входной файл	файл, содержащий входные данные;
сообщение	файл, который будет содержать созданное сообщение.

Примечание: Для того чтобы зашифровать данные «на себя», необходимо указать КПС своего сертификата.

Примеры:

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

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 -f "a:\Petr’s cert.p7b" test.txt test1.msg

Зашифровать содержимое файла «test.txt» в «test1.msg» (формат BASE64), используя сертификат из файла «a:\Petr’s cert.p7b».

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

-decr <КПС1> [-start] [-pin <пароль>|-askpin] <сообщение> <выходной файл>

КПС1	КПС получателя;
-start	открыть (запустить) полученный файл;
-askpin	запросить пароль ключевого контейнера с консоли;
-pin	задать пароль ключевого контейнера;
пароль	пароль к ключевому контейнеру;
сообщение	файл, содержащий сообщение;
выходной файл	файл, в который будут записаны данные из сообщения.

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

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.

2.4. Работа с пакетами файлов

Произвести хэширование содержимого файлов и записать результат в «имя_исходного_файла.hsh» можно с помощью команды

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

-dir	указать папку для файлов со значениями хэш-функции, иначе – текущая;
-provtype	указать тип криптопровайдера (N) (по умолчанию 75);
-provname	указать имя криптопровайдера (CSP);
-hashAlg	задать алгоритм хэширования
OID	OID алгоритма хэширования: 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 шестнадцатиричной строки;
маска файлов	стандартная маска хэшируемых файлов.

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

Пример:
cryptcp -hash -dir hashes -provtype 75 *.exe

Посчитать для всех файлов с расширением «exe» текущей папки значение хэш-функции и записать их в папку «hashes». При хэшировании использовать криптопровайдер по умолчанию для типа 75.

Проверить значение хэш-функций файла, созданное с помощью предыдущей команды, можно с помощью команды:

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

-dir	указать папку для файлов со значениями хэш-функции, иначе – текущая;
-provtype	указать тип криптопровайдера (**N**) (по умолчанию 75);
-provname	указать имя криптопровайдера (**<CSP>**);
-hex	значение хэш-функции в файле в виде шестнадцатиричной строки;
маска файлов	стандартная маска проверяемых файлов.

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

Пример:
cryptcp -vhash -dir c:\hashes -provtype 75 *.exe

Проверить для всех файлов с расширением «exe» текущей папки значение хэш-функции, эталонные значения хранятся в папке "c:\hashes". При хэшировании использовать криптопровайдер по умолчанию для типа 75.

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

[путь]cryptcp -signf [-dir <папка>] <КПС1> <маска файлов> [-cert] [-crl] [-der] [-strict] [-sd<URL>] [-ss<URL>] [-nostampcert] [-stampchaincheck] [-xlongtype1] [-cadesTSA<URL>] [-hashAlg <OID>] [-pin <пароль>|-askpin] [-display]

-dir	указать папку для файлов с подписями, иначе – текущая;
КПС1	КПС автора подписи;
-cert	добавлять в подписи сертификат отправителя;
-crl	добавлять в подписи список отозванных сертификатов;
-der	использовать формат DER вместо BASE64;
-strict	Использовать однозначное кодирование DER (а не BER)
-sd	добавить в подпись штамп времени на подписываемые данные (подписанный атрибут);
-ss	добавить в подпись штамп времени на подпись (неподписанный атрибут);
URL	адрес службы штампов в виде "http://..." (можно задать разные для опций -ss и -sd, но, если задан для одной из них, то используется и для второй);
-nostampcert	не требовать включения в штамп сертификата службы штампов времени (используется вместе с **-sd** и/или **-ss**);
-stampchaincheck	проверить цепочку сертификата в штампе времени
-xlongtype1	создавать подпись CAdES-X Long Type 1 (если данный параметр задан, параметр -ss будет проигнорирован)
-cadesTSA	служба штампов времени для подписи CAdES-X Long Type 1
URL	адрес службы штампов в виде "http://..."
-hashAlg	задать алгоритм хэширования
OID	OID алгоритма хэширования: 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
-askpin	запросить пароль ключевого контейнера с консоли;
-pin	задать пароль ключевого контейнера;
пароль	пароль к ключевому контейнеру;
-display	выводить информацию на экран средства доверенного отображения подписываемых данных;
маска файлов	стандартная маска подписываемых файлов.

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

Пример:

cryptcp -signf -dir \signs -uMyCerts -dn "Иванов Петр,ivanov@bank.ru" d:*.doc –sdhttp://cryptopro.ru/tsp/tsp.srf

Подписать содержимое всех файлов с расширением «doc» из корневой папки диска «d:», используя закрытый ключ, связанный с сертификатом хранилища «MyCerts» текущего пользователя, содержащим в поле «Субъект» («Subject») подстроки «Иванов Петр» и «ivanov@bank.ru», полученные подписи сохранить в папке «signs» в корне текущего диска. Кроме этого, получить штампы времени на каждый подписываемый файл и вложить их в соответствующие подписи.

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

[путь]cryptcp -vsignf [-dir <папка>] [-sd[<время>]] [-ss[<время>]] [-xlongtype1 | -nocades] <КПС> <маска файлов>

-dir указать папку с файлами, содержащими подписи, иначе – текущая;

КПС КПС автора подписи;
маска файлов стандартная маска проверяемых файлов.
-sd проверить штамп времени на подписанные данные (подписанный атрибут);
-ss проверить штамп времени на подпись (неподписанный атрибут);
время указывается в часах; если указано, то проверяет, чтобы штамп был сделан не ранее указанного количества часов назад от текущего момента;
-xlongtype1 проверить подпись CAdES-X Long Type 1, КПС будет проигнорирован

-nocades запретить использование вложенных в подпись доказательств

Пример:

cryptcp -vsignf -dir \signs -uMyCerts d:*.doc -sd24

Проверить все файлы с расширением «doc» из корневой папки диска «d:», используя созданные ранее подписи из папки «signs» в корне текущего диска. Поиск сертификата для проверки подписей искать в хранилище «MyCerts» текущего пользователя. Кроме этого, проверить штамп времени на подпись (неподписанный атрибут) и проверить, чтобы этот штамп был выдан не ранее, чем сутки назад.

Добавить подпись файла в 'исходный_файл.sgn' можно командой

[путь]cryptcp -addsignf [-dir <папка>] <КПС1> <маска файлов> [-cert] [-crl] [-der] [-sd<URL>] [-ss<URL>] [-nostampcert] [-stampchaincheck] [-xlongtype1] [-cadesTSA<URL>] [-pin <пароль>|-askpin]

-dir указать папку для файлов с подписями, иначе – текущая;
КПС1 КПС автора подписи;
-cert добавлять в подписи сертификат отправителя;
-crl добавлять в подписи список отозванных сертификатов;
-der использовать формат DER вместо BASE64;
-sd добавить в подпись штамп времени на подписываемые данные (подписанный атрибут);
-ss добавить в подпись штамп времени на подпись (неподписанный атрибут);
URL адрес службы штампов в виде "http://..." (можно задать разные для опций -ss и -sd, но, если задан для одной из них, то используется и для второй);
-nostampcert не требовать включения в штамп сертификата службы штампов времени
(используется вместе с -sd и/или -ss)
-stampchaincheck проверить цепочку сертификата в штампе времени
-xlongtype1 добавить подпись CAdES-X Long Type 1 (если данный параметр задан, параметр -ss будет проигнорирован)
-cadesTSA служба штампов времени для подписи CAdES-X Long Type 1
URL адрес службы штампов в виде "http://..."
-askpin запросить пароль ключевого контейнера из консоли
-pin задать пароль ключевого контейнера
пароль пароль к ключевому контейнеру
маска файлов маска для отбора подписываемых файлов

Пример: cryptcp -addsignf -dir /signs -uMy -dn "E=ivanov@test.ru, CN=Ivanov" /testdocuments/*.doc

Подписать все файлы с расширением «doc» из директории testdocuments с помощью сертификатов, находящихся в хранилище «My» текущего пользователя, удовлетворяющих следующим критериям: E=ivanov@test.ru, CN=Ivanov. Подписи добавить в файлы, расположенные в директории sings, соответствующие условию 'исходный_файл.sgn'

2.5. Работа с подписями

Подписать данные и создать сообщение можно следующим образом:

[путь]cryptcp -sign <КПС1> [-nocert] [-crl] [-der] [-strict] [-authattr <атрибут>]n раз [-attr <атрибут>]k раз [-sd<URL>] [-ss<URL>] [-nostampcert] [-stampchaincheck] [-xlongtype1] [-cadesTSA<URL>] [-hashAlg <OID>] [-pin <пароль>|-askpin] [-display] <входной файл> <сообщение>

КПС1	КПС автора подписи;
-nocert	не добавлять в сообщение сертификат отправителя;
-crl	добавление списка отозванных сертификатов;
-der	использовать формат DER вместо BASE64;
-strict	Использовать однозначное кодирование DER (а не BER);
-authattr	добавить подписанный атрибут в подпись;
-attr	добавить неподписанный атрибут в подпись;
атрибут	"<OID>,<файл с закодированным содержимым атрибута>"(пример: "1.2.3,attr.bin");
-sd	добавить в подпись штамп времени на подписываемые данные (подписанный атрибут);
-ss	добавить в подпись штамп времени на подпись (неподписанный атрибут);
URL	адрес службы штампов в виде "http://..." (можно задать разные для опций -ss и -sd, но, если задан для одной из них, то используется и для второй);
-nostampcert	не требовать включения в штамп сертификата службы штампов времени (используется вместе с **-sd** и/или **-ss**);
-stampchaincheck	проверить цепочку сертификата в штампе времени
-xlongtype1	проверить подпись CAdES-X Long Type 1, КПС будет проигнорирован
-cadesTSA	служба штампов времени для подписи CAdES-X Long Type 1
URL	адрес службы штампов в виде "http://..."
-hashAlg	задать алгоритм хэширования
OID	OID алгоритма хэширования: 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
-askpin	запросить пароль ключевого контейнера с консоли;
-pin	задать пароль ключевого контейнера;
пароль	пароль к ключевому контейнеру;
-display	выводить информацию на экран средства доверенного отображения подписываемых данных;
входной файл	файл, содержащий входные данные;
сообщение	файл, который будет содержать созданное сообщение.

Пример:

cryptcp -sign -mMy -dn Седов -q5 -nocert -crl -der test.txt test2.msg -sshttp://cryptopro.ru/tsp/tsp.srf

Подписать содержимое файла «test.txt» и создать подписанное сообщение «test2.msg» (в бинарном виде), не включающее в себя используемый сертификат, но включающее список отозванных сертификатов центра сертификации, выдавшего используемый сертификат. Кроме этого, получить штамп времени на созданную подпись и вложить ее в сообщение. Поиск используемого сертификата происходит следующим образом:

1. Находятся все сертификаты хранилища «Личные» текущего пользователя и локального компьютера.

2. Если их нашлось более пяти, то - ошибка, иначе пользователю будет предложено выбрать один из найденных сертификатов.

Добавить электронную подпись в сообщение можно с помощью вызова:

cryptcp -addsign <КПС1> [-nocert] [-crl] [-sd<URL>] [-ss<URL>] [-nostampcert] [-stampchaincheck] [-xlongtype1] [-cadesTSA<URL>] [-hashAlg <OID>] [-pin <пароль>|-askpin] [-authattr <атрибут>]n раз [-attr <атрибут>]k раз <сообщение>>

КПС1	КПС автора подписи;
-nocert	не добавлять в сообщение сертификат отправителя;
-crl	добавление списка отозванных сертификатов;
-authattr	добавить подписанный атрибут в подпись;
-attr	добавить неподписанный атрибут в подпись;
атрибут	"<OID>,<файл с закодированным содержимым атрибута>"(пример: "1.2.3,attr.bin");
-sd	добавить в подпись штамп времени на подписываемые данные (подписанный атрибут);
-ss	добавить в подпись штамп времени на подпись (неподписанный атрибут);
URL	адрес службы штампов в виде "http://..." (можно задать разные для опций -ss и -sd, но, если задан для одной из них, то используется и для второй);
-nostampcert	не требовать включения в штамп сертификата службы штампов времени (используется вместе с **-sd** и/или **-ss**);
-stampchaincheck	проверить цепочку сертификата в штампе времени
-xlongtype1	добавить подпись CAdES-X Long Type 1 (если данный параметр задан, параметр -ss будет проигнорирован)
-cadesTSA	служба штампов времени для подписи CAdES-X Long Type 1
URL	адрес службы штампов в виде "http://..."
-hashAlg	задать алгоритм хэширования
OID	OID алгоритма хэширования: 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
-askpin	запросить пароль ключевого контейнера с консоли;
-pin	задать пароль ключевого контейнера;
пароль	пароль к ключевому контейнеру;
сообщение	файл, содержащий сообщение.

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

Пример:
cryptcp -addsign -m -dn "Иванов Петр,ivanov@bank.ru" test.msg

Добавить в подписанное сообщение «test.msg» подпись, используя закрытый ключ, связанный с сертификатом хранилища «Личные» («My») локального компьютера, содержащим в поле «Субъект» («Subject») подстроки «Иванов Петр» и «ivanov@bank.ru». В добавленную подпись будет включен сертификат открытого ключа автора подписи.

Удалить электронную подпись из сообщения можно командой

-delsign <КПС1> <сообщение>

КПС1	КПС автора подписи;
сообщение	файл, содержащий сообщение.

Проверка электронной подписи

-verify [<КПС> | -verall] [-start] [-sd[<время>]] [-ss[<время>]] <сообщение> [<выходной файл>]

КПС	КПС авторов подписей;
-verall	проверять все подписи (иначе – только подписи авторов из КПС);
-start	открыть (запустить) полученный файл;
-sd	проверить штамп времени на подписанные данные (подписанный атрибут);
-ss	проверить штамп времени на подпись (неподписанный атрибут);
время	указывается в часах; если указано, то проверяет, чтобы штамп был сделан не ранее указанного количества часов назад от текущего момента;
-xlongtype1	проверить подпись CAdES-X Long Type 1, КПС будет проигнорирован
-nocades	запретить использование вложенных в подпись доказательств
сообщение	файл, содержащий сообщение;
выходной файл	файл, в который будут записаны данные из сообщения.

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

Примеры:

cryptcp -verify -dn ivanov@bank.ru test2.msg test2.txt

Проверить подпись сообщения "test2.msg", используя один из найденных сертификатов в хранилищах "Личные" ("My") и "Другие пользователи" ("AddressBook") текущего пользователя, содержащих в поле "Субъект" ("Subject") подстроку "ivanov@bank.ru" и записать содержимое подписанного сообщения в файл "test2.txt".

cryptcp -verify -sd3 test2.msg

Проверить все подписи сообщения "test2.msg", используя сертификаты, содержащиеся в сообщении. Если для какой-либо подписи в сообщении сертификат не удалось найти, то подпись проверена не будет. Кроме этого, проверить штамп времени на подписанные данные (подписанный атрибут) и проверить, чтобы этот штамп был выдан не ранее, чем три часа назад.

Добавить неподписанный атрибут в подпись можно с помощью команды

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

КПС1	КПС автора подписи;
-attr	добавить неподписанный атрибут в подпись;
атрибут	"<OID>,<файл с закодированным содержимым атрибута>"(пример: "1.2.3,attr.bin");
–сообщение	файл, содержащий сообщение.

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

2.6. Работа с сертификатами

Скопировать сертификаты в заданное хранилище можно с помощью команды

-copycert <КПС> [-{dm|du}[<имя>]|-df <файл> [-der]]

КПС	КПС, которые надо скопировать;

-dm копирование в хранилище компьютера (LOCAL_MACHINE);
-du* копирование в хранилище пользователя (CURRENT_USER);
имя название конечного хранилища (по умолчанию "My");
-df в качестве хранилища используется файл сертификата;
файл имя файла;
-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:

cryptcp -copycert -u -df a:\MyCerts.p7b

Копирует все сертификаты хранилища «Личные» («My») текущего пользователя в файл «a:\MyCerts.p7b» (в кодировке BASE64).

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

-CSPcert [-provtype <N>] [-provname <CSP>] [-cont <контейнер>] [-ku|-km] [-ex|-sg] [-{dm|du}[<имя>]|-df <файл> [-der]]

-provtype указать тип криптопровайдера (**N**) (по умолчанию 75);
-provname указать имя криптопровайдера (**CSP**);
-cont задать 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 копирование в хранилище компьютера (LOCAL_MACHINE);
-du* копирование в хранилище пользователя (CURRENT_USER);
имя название конечного хранилища (по умолчанию "My");
-df в качестве хранилища используется сообщение или файл сертификата;
файл имя файла;
-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:

cryptcp -CSPcert -km -cont WebServer -df a:\WebServer.cer -der

Копирует сертификат из ключевого контейнера «WebServer» криптопровайдера по умолчанию для типа 75 локального компьютера в файл "a:\WebServer.cer" (в кодировке DER).

Удалить сертификат из хранилища можно командой

-delcert <КПС> [-yes]

КПС КПС удаляемых сертификатов;
-yes автоматически отвечать на все вопросы «Да».

Пример:

cryptcp -delcert -m -dn OldServer

Удаляет все сертификаты хранилища «Личные» («My») локального компьютера, содержащие в поле «Subject» подстроку «OldServer».

2.7. Работа с запросами на сертификат

Команда для создания запроса сертификата и сохранение его в файле PKCS #10.

-creatrqst -dn <RDN> [-provtype <N>] [-provname <CSP>] [-SMIME] [-nokeygen|-exprt] [-keysize <n>] [-hashAlg <OID>] [-ex|-sg|-both] [-ku|-km] [-cont <имя>] [-silent] [-pin <пароль>|-askpin] [-certusage <OIDs>] [-der] [-ext <расширение>]n раз <имя файла>

RDN	список имен полей RDN (например: CN, O, E, L) и их значений вида:<ИмяПоля1>=<ЗначечиеПоля1>[,<ИмяПоля2>=<ЗначечиеПоля2> ...]
-provtype	указать тип криптопровайдера (**N**) (по умолчанию 75);
-provname	указать имя криптопровайдера (**CSP**);
-nokeygen	использовать существующие ключи из указанного контейнера;
-SMIME	включить возможности S/MIME (по умолчанию – нет; только Windows);
-exprt	пометить ключи как экспортируемые;
-keysize	установить длину ключа (n);
-hashAlg	задать алгоритм хэширования
OID	OID алгоритма хэширования: 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 ключи для обмена зашифрованными данными;
-sg	создать/использовать ключи только для работы с подписями;
-both*	создать/использовать оба типа ключей;
-ku*	использовать контейнер пользователя (CURRENT_USER);
-km	использовать контейнер компьютера (LOCAL_MACHINE);
-cont	задать имя ключевого **контейнера** (если задана опция -nokeygen и не задана опция -cont – выбор из списка);
-silent	генерация ключа без пользовательского интерфейса криптопровайдера;
-askpin	запрашивать пароль при создании ключевого контейнера с консоли (только UNIX);
-pin	установить пароль при создании ключевого контейнера (только UNIX);
пароль	пароль к ключевому контейнеру (только UNIX);
-certusage	задать назначения сертификата (**OIDs**). Если назначений несколько, то их необходимо указать через запятую (например, ”1.3.6.1.5.5.7.3.4, 1.3.6.1.5.5.7.3.2”);
-requestlic	запросить сертификат, содержащий расширение с лицензией на КриптоПро CSP. УЦ должен быть настроен на выдачу таких сертификатов;
-der	использовать формат DER вместо BASE64;
-ext	добавить расширение к запросу;
расширение	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:

cryptcp -creatrqst c:\request.der -provtype 75 -cont Ivanov -dn "E=ivanov@bank.ru, CN=Иванов Петр" -both –ku -provname "Crypto-Pro GOST R 34.10-2001 Cryptographic Service Provider"

Создать запрос на субъект "E=ivanov@bank.ru,CN=Иванов Петр", используя открытый ключ, сгенерированный в контейнере "Ivanov" текущего пользователя криптопровайдером "Crypto-Pro

GOST R 34.10-2001 Cryptographic Service Provider" (тип - 75) и сохранить его в файл c:\request.der в кодировке Base64. Назначения ключа - подпись и шифрование.

Установка сертификата из файла PKCS #7 или файла сертификата.

-instcert [-provtype <N>] [-provname <CSP>] [-cont <имя>] [-ku|-km] [-{dm|du}[<имя>]] [-noCSP] [-pin <пароль>|-askpin] <имя файла>

-provtype	указать тип криптопровайдера (**N**) (по умолчанию 75);
-provname	указать имя криптопровайдера (**CSP**);
-cont	задать имя ключевого **контейнера** (по умолчанию выбор из списка);
-ku*	использовать контейнер пользователя (CURRENT_USER);
-km	использовать контейнер компьютера (LOCAL_MACHINE);
-dm	установка в хранилище компьютера (LOCAL_MACHINE);
-du*	установка в хранилище пользователя (CURRENT_USER);
имя	название конечного хранилища для установки (по умолчанию «My»);
-noCSP	не сохранять сертификат в контейнере криптопровайдера;
-askpin	запросить пароль ключевого контейнера с консоли (только UNIX);
-pin	задать пароль ключевого контейнера (только UNIX);
пароль	пароль к ключевому контейнеру (только UNIX);
-enable-install-root	не запрашивать разрешение на установку корневого сертификата в хранилище «Доверенные корневые центры» (Root) (только UNIX);
имя файла	имя файла, содержащего сертификат

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").

Извлечение информации о настройках параметров учетных записей пользователя на УЦ

-listDN [{-CA <адрес ЦС>}|{-CPCA <адрес ЦС>} |{-CPCA20 <адрес ЦС>}]

-CPCA	указать адрес веб интерфейса КриптоПро УЦ;
адрес УЦ	вида \"http://xxx.yyy/zzz\" или \"https://xxx.yyy/zzz\";
-CPCA20	указать адрес веб интерфейса КриптоПро УЦ версии 2.0;
адрес УЦ	вида \"https://xxx.yyy/zzz/{folder}\";
folder	обозначает GUID папки УЦ или путь папки в иерархии папок УЦ, при этом разделителем имен папок в пути является символ '|'

Регистрация пользователя на УЦ

-creatuser [-CPCA <адрес УЦ>] [-field <ID поля = значение>]n раз

-CPCA	указать адрес веб интерфейса КриптоПро УЦ;
адрес УЦ	вида \"http://xxx.yyy/zzz\" или \"https://xxx.yyy/zzz\";
-CPCA20	указать адрес веб интерфейса КриптоПро УЦ версии 2.0;
адрес УЦ	вида \"https://xxx.yyy/zzz/folder\"; folder обозначает GUID папки УЦ или путь папки в иерархии папок УЦ, при этом разделителем имен папок в пути является символ '|'
-field	добавить поле в DN регистрируемого пользователя
ID поля	идентификатор поля DN. Список идентификаторов можно посмотреть

командой cryptcp –listDN

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

Проверить, зарегистрирован ли пользователь на УЦ

-checkreg -token <ID маркера> -tpassword <пароль> [-CPCA <адрес УЦ>]

-token	задать маркер временного доступа для проверки статуса
-tpassword	задать пароль к маркеру временного доступа
-CPCA	указать адрес веб интерфейса КриптоПро УЦ, иначе это адрес "CP CSP Test CA": https://cryptopro.ru/ui
адрес УЦ	вида "http://xxx.yyy/zzz" или "https://xxx.yyy/zzz"
-CPCA20	указать адрес веб интерфейса КриптоПро УЦ версии 2.0, иначе это адрес "CP CSP Test CA": https://cryptopro.ru/ui
адрес УЦ	вида "https://xxx.yyy/zzz/

Создать запрос на сертификат, отправить его в центр сертификации, получить выписанный сертификат и установить его.

[путь]cryptcp -creatcert -rdn <RDN> [-provtype <N>] [-provname <CSP>] [-SMIME] [-nokeygen|-exprt] [-keysize <n>] [-hashAlg <OID>] [-{ex|sg|both}] [-cont <имя>] [-ku|-km] [-certusage <OIDs>] [{-CA <адрес ЦС>}|{-CPCA <адрес УЦ>}] [-requestlic] [{-token <ID токена> -tpassword <пароль>}| -clientcert КПС1] [-{dm|du}[<имя>]] [-noCSP] [-silent] [-pin <пароль>|-askpin] [-keysize <N>] [-FileID <Имя файла>] [-ext <расширение>]n раз [-enable-install-root]

RDN	список имен полей RDN (например: CN, O, E, L) и их значений вида: <ИмяПоля1>=<ЗначениеПоля1>[,<ИмяПоля2>=<ЗначениеПоля2>...]
-provtype	указать тип криптопровайдера (**N**) (по умолчанию 75);
-provname	указать имя криптопровайдера (**CSP**);
-SMIME	включить возможности S/MIME (по умолчанию – нет; только Windows);
-nokeygen	использовать существующие ключи из указанного контейнера;
-exprt	пометить ключи как экспортируемые;
-**keysize**	установить длину ключа (n);
-hashAlg	задать алгоритм хэширования
OID	OID алгоритма хэширования: 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 ключи для обмена зашифрованными данными;
-sg	создать/использовать ключи только для работы с подписями;
-both*	создать/использовать оба типа ключей;
-ku*	использовать контейнер пользователя (CURRENT_USER);
-km	использовать контейнер компьютера (LOCAL_MACHINE);
-cont	задать имя ключевого **контейнера** (по умолчанию выбор из списка);
-certusage	задать назначения сертификата (**OIDs**). Если назначений несколько, то их нужно указать через запятую (например, ”1.3.6.1.5.5.7.3.4,1.3.6.1.5.5.7.3.2”);
-CA	указать адрес центра сертификации, иначе это адрес “CP CSP Test CA”;
адрес ЦС	вида “http://xxx.yyy/zzz” или “\\сервер\имяЦС” (см. ”Системные требования”);
-CPCA	указать адрес веб интерфейса КриптоПро УЦ;
адрес УЦ	вида \"http://xxx.yyy/zzz\" или \"https://xxx.yyy/zzz\";
-CPCA20	указать адрес веб интерфейса КриптоПро УЦ версии 2.0;

адрес УЦ	вида \"https://xxx.yyy/zzz\";
-tmpl	задать шаблон запрашиваемого сертификата (только УЦ КриптоПро версии 2.0);
имя	название или OID шаблона (только УЦ КриптоПро версии 2.0)
-token	использовать маркер временного доступа для аутентификации на УЦ КриптоПро;
-tpassword	задать пароль к маркеру временного доступа;
<КПС1>	использовать сертификат для аутентификации на УЦ КриптоПро (только для Unix);
-dm	установка в хранилище компьютера (LOCAL_MACHINE);
-du*	установка в хранилище пользователя (CURRENT_USER);
имя	название конечного хранилища для установки (по умолчанию “My”);
-noCSP	не сохранять сертификат в контейнере криптопровайдера;
-silent	генерация ключа без пользовательского интерфейса криптопровайдера;
-askpin	запрашивать пароль при создании ключевого контейнера с консоли (только UNIX);
-pin	установить пароль при создании ключевого контейнера (только UNIX);
пароль	пароль к ключевому контейнеру (только UNIX);
-requestlic	запросить сертификат, содержащий расширение с лицензией на КриптоПро CSP. УЦ должен быть настроен на выдачу таких сертификатов;
-FileID	имя файла, используемого для записи идентификатора запроса в случае “отложенной выдачи” сертификата (см. **-pendcert**). Если файл не указан, то идентификатор будет выведен на экран.
-enable-install-root	не запрашивать разрешение на установку корневого сертификата в хранилище “Доверенные корневые центры” (Root) (только UNIX);
-ext	добавить расширение к запросу;
расширение	имя файла с закодированным расширением (BASE64 или DER);

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”).

Проверить, не выпущен ли сертификат, запрос на который был отправлен ранее, получить выписанный сертификат и установить его.

-pendcert [-provtype <N>] [-provname <CSP>] [-cont <имя>] [-ku|-km]
[{-CA <адрес ЦС>}|{-CPCA <адрес ЦС>} |{-CPCA20 <адрес ЦС>}]
[{-token <ID токена> -tpassword <пароль>} | -clientcert КПС1]
[-{dm|du}[<имя>]] [-noCSP] [-FileID <Имя файла>]
[-pin <пароль>|-askpin]

-provtype	указать тип криптопровайдера (**N**) (по умолчанию 75);
-provname	указать имя криптопровайдера (**CSP**);
-cont	задать имя ключевого **контейнера** (по умолчанию выбор из списка);
-ku*	использовать контейнер пользователя (CURRENT_USER);
-km	использовать контейнер компьютера (LOCAL_MACHINE);
-CA	указать адрес центра сертификации, иначе это адрес “CP CSP Test CA”;
адрес ЦС	вида “http://xxx.yyy/zzz” или “\\сервер\имяЦС” (см. ”Системные

	требования”);
-CPCA	указать адрес веб интерфейса КриптоПро УЦ;
адрес УЦ	вида \"http://xxx.yyy/zzz\" или \"https://xxx.yyy/zzz\";
-CPCA20	указать адрес веб интерфейса КриптоПро УЦ версии 2.0;
адрес УЦ	вида \"https://xxx.yyy/zzz/\";
-token	использовать маркер временного доступа для аутентификации на УЦ КриптоПро;
-tpassword	задать пароль к маркеру временного доступа;
<КПС1>	использовать сертификат для аутентификации на УЦ КриптоПро (только для Unix);
-dm	установка в хранилище компьютера (LOCAL_MACHINE);
-du*	установка в хранилище пользователя (CURRENT_USER);
имя	название конечного хранилища для установки (по умолчанию “My”);
-noCSP	не сохранять сертификат в контейнере криптопровайдера;
-FileID	имя файла, содержащего идентификатор запроса. Если не файл не указан, то идентификатор нужно будет ввести вручную.
-askpin	запросить пароль ключевого контейнера с консоли (только UNIX);
-pin	задать пароль ключевого контейнера (только UNIX);
пароль	пароль к ключевому контейнеру (только UNIX);
-enable-install-root	не запрашивать разрешение на установку корневого сертификата в хранилище “Доверенные корневые центры” (Root) (только UNIX);

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”).

2.8. Команда для работы с серийным номером лицензии (только для Windows)

Сохранить/показать серийный номер лицензии.

-sn [<серийный номер>]

серийный номер	серийный номер, который необходимо сохранить (можно указывать как с разделителями, так и без них).

Примечание: Для того чтобы посмотреть сохраненный серийный номер, достаточно указать команду -sn без параметра. В операционных системах семейства UNIX команда не используется.

Пример:
cryptcp -sn P020G-Q0010-A5000-01UXA-XUFFD

Сохраняет указанный серийный номер лицензии на компьютере.

2.9. Усовершенствованная электронная подпись

Приложение командной строки поддерживает возможность создания улучшенной электронной подписи, соответствующей стандарту CAdES (см. RFC 5126 "CMS Advanced Electronic Signatures (CAdES)"). 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 RFC 3161 "Internet X.509 Public Key Infrastructure Time-Stamp Protocol (TSP)".

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
-cadesTSA	указывается служба штампов времени для подписи CAdES-X Long Type 1
-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 управление лицензиями КриптоПро PKI.

Пример:

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

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

3. Возвращаемые коды ошибок

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 сертификата
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 секретным ключом

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.