СИСТЕМНЫЕ ПРОГРАММЫ ДЛЯ РАБОТЫ С СЕРТИФИКАТАМИ

1. Создание сертификатов

Для создания сертификата требуется иметь секретный ключ его издателя. В состав операционной системы Windows входит сертификат издателя по умолчанию (например, Root Agency) и связанный с ним секретный ключ.

Самоподписанные (self-signed) сертификаты могут использоваться как в тестовых целях, так и в качестве доверенных корневых сертификатов. Подпись под таким сертификатом вычисляется с помощью секретного ключа, связанного с открытым ключом из создаваемого сертификата. Доверенные корневые сертификаты могут затем использоваться для удостоверения новых сертификатов.

Для создания самоподписанных сертификатов, а также сертификатов, удостоверенных с их помощью или с помощью имеющихся сертификатов издателей, может использоваться системная программа MakeCert, которая представляет собой утилиту командной строки. Формат строки вызова этой системной программы следующий:

MakeCert [*базовые опции* | *расширенные опции*] [*имя выходного файла*]

Имя выходного файла может быть опущено, если создаваемый сертификат не должен записываться в файл.

Опции, которые могут быть указаны при вызове утилиты MakeCert, разделяются на три группы:

- базовые опции, управляющие созданием и хранением созданного сертификата;
- расширенные опции, применимые к свойствам создаваемого сертификата, *сертификатам издателей* и связанным с ними секретным ключам и их хранению.

Большинство опций программы MakeCert доступны при использовании обозревателя Internet Explorer версии 4.0 и выше. В табл. 1 приведено описание базовых опций программы MakeCert.

Табл. 1

Опция	Описание опции
/n *имя*	Имя владельца сертификата в формате X.500 (например, “CN=*имя*”)
/pe	Созданный личный ключ отмечается как экспортируемый
/sk *имя*	Имя существующего или создаваемого контейнера ключей, который будет использован для хранения секретного ключа (если не указаны ни /sk, ни /sv опция, то используется имя контейнера по умолчанию, например JoeSoft)
/sr *раздел*	Раздел реестра, используемый для размещения создаваемого сертификата (LocalMachine или CurrentUser, по умолчанию)

/ss *имя*	Имя хранилища для создаваемого сертификата
/# *номер*	Серийный номер создаваемого сертификата, от 1 до 2^{31}-1 (по умолчанию генерируется утилитой MakeCert с гарантией уникальности)
/$ *тип*	Тип удостоверяющего центра (commercial – для удостоверения издателей коммерческого программного обеспечения или individual – для удостоверения издателей индивидуального программного обеспечения)

В табл. 2 приведено описание расширенных опций утилиты MakeCert, предназначенных для управления свойствами создаваемого сертификата, сертификатами и секретными ключами издателей.

Табл. 2

Опция	**Описание опции**
/a *алгоритм*	Алгоритм хеширования (sha1 или md5, по умолчанию)
/b *дата*	Дата начала действия сертификата (в формате mm/dd/yyyy), по умолчанию – дата создания сертификата
/cy *тип*	Тип сертификата (end – пользовательский, authority – удостоверяющего центра)
/e *дата*	Дата окончания действия сертификата (по умолчанию 12/31/2039)
/eku *OID1,OID2…*	Список идентификаторов объектов для назначений сертификата (по умолчанию – все назначения)
/h *длина*	Максимальная длина цепочки сертификации
/ic *строка*	Имя pvk-файла с секретным ключом издателя
/ik *строка*	Имя контейнера ключей издателя (имя по умолчанию зависит от версии операционной системы)
/iky *тип*	Тип создаваемого ключа издателя (аналогично опции /sky)
/in *имя*	Имя издателя сертификата (утилита MakeCert будет искать сертификат, чье имя включает заданное этой опцией)
/ip *имя*	Криптопровайдер для издателя
/ir *раздел*	Раздел реестра для хранения сертификата издателя (LocalMachine или CurrentUser, по умолчанию)
/is *имя*	Имя хранилища сертификатов, содержащего сертификат издателя и связанный с ним секретный ключ (если таких сертификатов несколько, то требуется его явная спецификация с помощью опций /ic или /in; если не находится уникального сертификата, то утилита MakeCert завершается с ошибкой)
/iv *имя файла*	Имя pvk-файла с секретным ключом издателя (имя по умолчанию зависит от версии операционной системы)
/iy *тип*	Тип криптопровайдера для издателя (по умолчанию PROV_RSA_FULL)
/l *ссылка*	Ссылка (например, URL) на ресурс, определяющий политику использования

/len *число*	Длина в битах генерируемого ключа
/m *число*	Срок действия сертификата в месяцах
/nscp	Добавление расширения для клиента аутентификации Netscape
/r	Создание самоподписанного сертификата
/sc *имя файла*	Имя файла с уже существующим сертификатом
/sky *тип*	Тип создаваемого ключа асимметричного шифрования (signature (1) – ключ ЭЦП, exchange (2) – ключ обмена, 3 – ключ ЭЦП и обмена); если опция не задана, то тип определяется типом используемых файла или контейнера ключей (если в контейнере есть разные ключи, то вначале MakeCert пытается создать ключ ЭЦП, а затем – ключ обмена)
/sp *имя*	Имя криптопровайдера (если опция не задана, то используется криптопровайдер по умолчанию)
/sv *имя файла*	Имя файла с расширением .pvk для хранения секретного ключа
/sy *тип*	Тип криптопровайдера (по умолчанию PROV_RSA_FULL)

При вызове утилиты MakeCert без указания опций и имени выходного файла программа завершается с ошибкой и выводом информации об основных ее базовых опциях:

```
Error: Please either specify the outputCertificateFile or -ss option
Usage: MakeCert [ basic|extended options] [outputCertificateFile]
Basic Options
 -sk  <keyName>      Subject's key container name; To be created if not present
 -pe                 Mark generated private key as exportable
 -ss  <store>        Subject's certificate store name that stores the output
                     certificate
 -sr  <location>     Subject's certificate store location.
                     <CurrentUser|LocalMachine>.  Default to 'CurrentUser'
 -#   <number>       Serial Number from 1 to 2^31-1.  Default to be unique
 -$   <authority>    The signing authority of the certificate
                     <individual|commercial>
 -n   <X509name>     Certificate subject X500 name (eg: CN=Fred Dews)
 -?                  Return a list of basic options
 -!                  Return a list of extended options
```

Рассмотрим примеры использования утилиты MakeCert. В первом примере создается сертификат, изданный издателем по умолчанию (например, Root Agency). Созданный сертификат сохраняется в файле. Утилита вызывается в режиме командной строки (сообщение Succeeded подтверждает успешность создания сертификата):

```
makecert mynew.cer
Succeeded
```

В следующем примере созданный и подписанный издателем по умолчанию сертификат помещается в хранилище myNewStore:

```
MakeCert /ss myNewStore
```

В третьем примере создается сертификат, подписанный издателем по умолчанию. Этот сертификат сохраняется в хранилище myNewStore и в файле myNew.cer, а созданный секретный ключ помещается в файл myNew.pvk:

```
MakeCert /sv myNew.pvk /ss myNewStore myNew.cer
```

При создании файла с секретным ключом будет запрошен пароль для генерации ключа шифрования экспортируемого секретного ключа. После ввода и подтверждения этого пароля для доступа к секретному ключу пользователя потребуется вводить пароль.

В следующем примере создаются подписанный издателем по умолчанию сертификат и контейнер ключей с именем myNewKey для хранения созданных ключей, а созданный сертификат сохраняется и в хранилище myNewStore, и в файле myNew.cer:

```
MakeCert /sk myNewKey /ss myNewStore myNew.cer
```

В пятом примере создаются секретный ключ и подписанный издателем по умолчанию сертификат, который сохраняется в хранилище myNewStore, а затем используется для удостоверения еще одного нового сертификата, сохраняемого в другом хранилище anotherStore. В этом примере утилита MakeCert завершается с ошибкой (в хранилище myNewStore было обнаружено более одного сертификата издателя):

```
MakeCert /sk myNewKey /ss myNewStore
MakeCert /is myNewStore /ss anotherStore
Error: There are more than one matching certificate in the issuer's myNewStore cert store
Failed
```

В следующем примере создается подписанный издателем по умолчанию сертификат, который сохраняется в хранилище My, а затем используется для удостоверения вновь созданного сертификата. Выбор сертификата издателя в хранилище My осуществляется по имение издателя:

```
MakeCert /sk myNewKey /n "CN=XXZZYY" /ss my
MakeCert /is my /in "XXZZYY" /ss anotherStore
```

В седьмом примере создается подписанный издателем по умолчанию сертификат, который сохраняется в хранилище My и в файле myNew.cer. Этот сертификат используется далее для удостоверения вновь созданного сертификата (в качестве сертификата издателя выбирается первый подходящий сертификат на основе указанного файла):

```
MakeCert /sk myNewKey /n "CN=XXZZYY" /ss my myNew.cer
MakeCert /is my /ic myNew.cer /ss anotherStore
```

В следующем примере создается самоподписанный сертификат, который помещается в хранилище myNewRoot:

```
MakeCert /sk myNewRootKey /r /ss myNewRoot
```

В последнем примере создается самоподписанный сертификат, который сохраняется в системном хранилище CA и в файле myNewRoot.cer. Затем этот сертификат используется для удостоверения вновь созданного сертификата, помещаемого в хранилище myNewSign:

MakeCert /sk myNewRootKey /r /ss ca myNewRoot.cer

MakeCert /is ca /ic myNewRoot.cer /ss myNewSign

Для использования утилиты MakeCert (и других утилит, рассматриваемых в этой главе, кроме программы selfcert.exe) необходимо установить пакет Microsoft.NET Framework Software Development Kit (SDK) (или Platform SDK for Windows Server 2003 SP1 Full Download, или Microsoft Visual Studio 2005), которые могут быть, например, загружены с сайта корпорации Microsoft.

Самоподписанный сертификат может быть также создан с помощью программы selfcert.exe, входящей в комплект поставки пакета Microsoft Office XP и более поздних версий (в версии Microsoft Office 2003 вызов этой программы возможен через меню Пуск | Microsoft Office | Средства Microsoft Office | Цифровой сертификат для проектов VBA).

При создании самоподписанного сертификата (вместе с соответствующим ему секретным ключом ЭЦП) программа Selfcert.exe запросит имя владельца сертификата, а после успешного завершения процедуры создания выдаст соответствующее сообщение.

Созданный таким образом сертификат будет помещен в хранилище My. Сертификат будет предназначен для подписания кода, а срок его действия будет составлять 6 лет (начиная с первого дня текущего года). Для хранения ключей будет создан контейнер SelfSignedCerts.

Чтобы самоподписанный сертификат мог использоваться для удостоверения сообщений и других сертификатов он должен быть экспортирован в хранилище доверенных корневых сертификатов.

Утилита Cert2SPC создает тестовый сертификат издателя программного обеспечения (Software Publisher Certificate, SPC), используя уже существующие сертификаты. Эта утилита может объединить несколько сертификатов в одном закодированном документе формата PKCS #7. Сертификат SPC для реального использования должен быть получен в одном из удостоверяющих центров.

Формат строки вызова утилиты Cert2SPC:

Cert2SPC *cert1.cer cert2.cer. . .certN.cer output.spc*

Здесь cert1.cer, cert2.cer, . . ., certN.cer означают имена файлов с сертификатами, включаемыми в SPC (расширение .cer обязательно), а output.spc – имя выходного файла с сертификатом SPC (расширение .spc обязательно).

В приводимом далее примере сертификат из файла MyCert.cer преобразуется в SPC, который помещается в файл MyCert.spc.

Cert2SPC MyCert.cer MyCert.spc

Для преобразования файлов с секретным (личным, закрытым) ключом в формате PVK и сертификатом открытого ключа издателя программного

обеспечения в формате SPC в файл обмена персональной информацией формата PFX может использоваться утилита командной строки pvk2pfx. Формат строки вызова этой утилиты:

pvk2pfx /pvk *pvk-файл* [/pi *pvk-пароль*] /spc *spc-файл*
[/pfx *pfx-файл* [/po *pfx-пароль*] [/f]]

Если не указана опция pi, то пароль для расшифрования секретного ключа из pvk-файла будет запрашиваться в специальном диалоговом окне. Если не указана опция pfx, то начнется диалог с мастером экспорта сертификата, в котором и потребуется задать имя и месторасположение pfx-файла и пароль для шифрования экспортируемого секретного ключа (см. раздел 4), а опции po и f при этом игнорируются. Если опция pfx задана, но не указана опция po, то пароль для доступа к pfx-файлу будет совпадать с паролем для доступа к pvk-файлу, заданным опцией pi. Если указаны опции pfx и f, то в случае существования pfx-файла он будет перезаписан.

2. Создание списка доверенных сертификатов

Утилита MakeCTL предназначена для создания списков доверенных сертификатов (CTL). Созданный список кодируется и сохраняется в хранилище сертификатов или файле.

Входом утилиты MakeCTL является массив хранилищ сертификатов. Вычисляются хеш-значения всех сертификатов в этих хранилищах, которые и включаются в CTL.

Хранилища сертификатов могут быть заданы следующими способами:

- сохраненным ранее файлом хранилища;
- файлом в формате PKCS #7;
- файлом с закодированным сертификатом;
- именем системного хранилища.

Формат командной строки при вызове утилиты MakeCTL:

MakeCTL [/u *subjectUsageID*] [/s [/r *registryLocation*]]
хранилище сертификатов 1 [/s [/r *registryLocation*]]
хранилище сертификатов 2 … [/s [/r *registryLocation*]]
хранилище сертификатов N имя выходного файла.stl

Здесь subjectUsageID – идентификатор объекта для назначения создаваемого CTL (по умолчанию этот список состоит из сертификатов корневых удостоверяющих центров, предназначенных для подписания кода, что задается константой szOID_TRUSTED_CODESIGNING_CA_LIST, определенной в файле Wintrust.h как 1.3.6.1.4.1.311.2.2.1), registryLocation – указатель на размещение в реестре системного хранилища сертификатов (по умолчанию currentUser, т.е. используется раздел HKEY_CURRENT_USER, но возможно и задание localMachine для указания на раздел HKEY_LOCAL_MACHINE).

Опция /s указывает на то, что используется системное хранилище сертификатов. Может быть дополнительно указана опция /? для получения

информации о синтаксисе командной строки и возможных опциях при вызове утилиты MakeCTL.

Перед своим использованием закодированный файл с CTL должен быть подписан с помощью утилиты SignTool с командой sign. После этого CTL может быть помещен в системное хранилище Trust или Root с помощью утилиты CertMgr. Если идентификатор объекта для назначения CTL совпадает с szOID_TRUSTED_CODESIGNING_CA_LIST (значением по умолчанию), то все файлы, подписанные с помощью сертификатов из CTL, могут быть аутентифицированы с помощью утилиты SignTool с командой verify.

Рассмотрим два примера использования утилиты MakeCTL. В первом примере создаваемый список будет включать все сертификаты из системного хранилища Root, а CTL будет помещен в файл output.stl:

MakeCTL /s root output.stl

Во втором примере в CTL включаются три сертификата из файлов с расширением .cer:

MakeCTL one.cer two.cer three.cer output.stl

Проводник Windows может проверять CTL: двойной щелчок на stl-файле приведет к открытию соответствующего диалога. На вкладке «Общие» этого диалога приводятся сведения о результатах проверки ЭЦП под CTL, его версии, назначении сертификатов в этом списке и других характеристиках. На вкладке «Список доверия» содержится информация о включенных в список CTL сертификатах, их основных элементах и значении ЭЦП.

Если файл со списком доверенных сертификатов имеет неправильный формат или подпись, то диалог списка доверия сертификатов не открывается.

При использовании совместно с обозревателем Internet Explorer 5.0 и старше вызов утилиты MakeCTL в командной строке без параметров приводит к запуску мастера списков доверия сертификатов.

На первом шаге работы с мастером списка доверия сертификатов приводятся сведения о мастере и назначении CTL. На втором шаге пользователь может ввести префикс, характеризующий создаваемый CTL, и срок его действия в месяцах или днях. Обязательным на этом шаге является определение назначений для создаваемого списка. Помимо уже перечисленных мастером назначений можно выбрать и другое, указав его идентификатор объекта.

На третьем шаге взаимодействия с мастером пользователь должен добавить в создаваемый CTL сертификаты из хранилищ или из файла, причем в список сертификатов для возможного выбора будут включены только сертификаты с соответствующими назначениями, выбранными на предыдущем шаге.

На четвертом шаге работы с мастером нужно выбрать размещение создаваемого CTL – хранилище сертификатов или файл.

На пятом шаге взаимодействия с мастером задаются понятное имя создаваемого списка и его описание. На шестом шаге пользователю

предлагается подтвердить все характеристики создаваемого CTL. Если создание списка доверенных сертификатов завершилось успешно, будет выведено соответствующее сообщение.

3. Вычисление и проверка электронной цифровой подписи

Утилита SignTool с командой sign предназначена для вычисления электронной цифровой подписи под файлом. Если файл уже содержит ЭЦП, то подпись будет вычислена заново. Формат командной строки при вызове утилиты SignTool с командой sign:

SignTool sign [*опции*] *имя файла*

Утилита SignTool с командой sign поддерживает три группы опций:

- опции, влияющие на выбор сертификата (табл. 3);
- опции, относящиеся к секретному ключу (табл. 4);
- опции, относящиеся к создаваемой ЭЦП (табл. 5);
- другие опции (табл. 6).

Табл. 3

Опция	Описание опции
/a	Выбирается лучший из подходящих сертификатов (иначе ожидается, что существует один подходящий сертификат)
/c *имя*	Имя шаблона сертификата
/f *имя*	Имя файла с сертификатом (для PFX-файла, защищенного паролем требуется опция /p, а если файл не содержит личный ключ, то могут использоваться опции /csp и /k)
/i *имя*	Имя или часть имени издателя сертификата подписи
/j *имя*	Имя файла с DLL, возвращающей массив атрибутов подписи
/jp *параметр*	Параметр (только один) для передачи в определенную предыдущей опцией DLL
/n *имя*	Имя или часть имени владельца сертификата подписи
/p *строка*	Пароль для PFX-файла с личным ключом
/r *имя*	Имя владельца корневого сертификата, удостоверяющего сертификат подписи
/s *имя*	Хранилище сертификатов, содержащее сертификат и секретный ключ создателя ЭЦП (по умолчанию My)
/sm	Для поиска сертификата подписи используется хранилище в разделе реестра HKEY_LOCAL_MACHINE (иначе в HKEY_CURRENT_USER)
/sha1 *отпечаток*	Хеш-значение сертификата создателя ЭЦП
/u *OID или строка*	Расширенное назначение ключа ЭЦП (по умолчанию “Code Signing” (1.3.6.1.5.5.7.3.3), т.е. подписание кода)
/uw	Назначение ключа ЭЦП − “Windows System Component Verification” (1.3.6.1.4.1.311.10.3.6), т.е. проверка компонент Windows)

Табл. 4

Опция	Описание опции
/csp *имя*	Имя криптопровайдера, содержащего контейнер ключей с личным ключом подписи
/k *имя*	Имя контейнера ключей с секретным ключом

Табл. 5

Опция	Описание опции
/d *строка*	Описание подписываемого файла
/du *URL*	Адрес в сети Интернет с информацией о подписываемом файле
/t *URL*	Адрес в сети Интернет сервера отметок времени (если эта опция отсутствует, то отметка времени не включается в подписываемый файл; если получение отметки времени завершается с ошибкой, то генерируется предупреждение)

Табл. 6

Опция	Описание опции
/q	При успешном завершении не генерируется никаких сообщений, а при ошибке – минимальное количество сообщений
/v	Выводится максимально возможная информация как при успехе, так и при ошибке

Утилита SignTool с командой timestamp используется для добавления отметки времени в ранее подписанный файл. В этом случае допускаются опции /t (обязательная опция), /q и /v.

В следующем примере вызова утилиты SignTool вычисляется ЭЦП и ставится отметка времени на файл MyControl.exe (будет автоматически выбран наилучший из подходящих сертификатов).

```
SignTool sign /a /du http://example.microsoft.com
/t http://timestamp.verisign.com/scripts/timstamp.dll MyControl.exe
```

Второй пример отличается от первого тем, что секретный ключ берется из файла My.pfx.

```
SignTool sign /f My.pfx /p password
/du http://example.microsoft.com
/t http://timestamp.verisign.com/scripts/timstamp.dll MyControl.exe
```

В обоих случаях сертификат подписи (из хранилища или из файла) встраивается в файл MyControl.exe, а к вычисленной ЭЦП добавляется отметка времени.

В следующем примере сертификат создателя ЭЦП ищется по его имени (MyCert) в хранилище сертификатов по умолчанию. Подписанный программный файл получает понятное имя My Control.

```
SignTool sign /n "myCert" /d "My Control"
/du http://example.microsoft.com
/t http://timestamp.verisign.com/scripts/timstamp.dll MyControl.exe
```

В заключительном примере файл подписывается и снабжается отметкой времени с помощью единственного сертификата, содержащегося в хранилище DemoCert.

```
SignTool sign /s "DemoCert" /d "My control"
```

/du http://example.microsoft.com
/t http://timestamp.verisign.com/scripts/timstamp.dll MyControl.exe

При вызове утилиты SignTool signwizard без опций на компьютере с установленным обозревателем Internet Explorer версии 5.0 и старше начинается диалог с мастером создания цифровой подписи. На первом шаге работы с мастером пользователю отображаются сведения о назначении мастера и ЭЦП.

На втором шаге взаимодействия с мастером пользователь должен задать имя подписываемого файла. Третий шаг диалога с мастером позволяет выбрать тип создаваемой подписи. Если пользователь выбирает особый тип подписи, то он сможет явно задать ряд ее параметров:

- указать сертификат, который вместе с соответствующим секретным ключом будет использован для вычисления ЭЦП;
- указать файл или контейнер ключей криптопровайдера, содержащий секретный ключ для вычисления ЭЦП;
- выбрать алгоритм хеширования, который будет использован для вычисления ЭЦП;
- определить, какие дополнительные сертификаты должны быть включены в ЭЦП.

После определения параметров ЭЦП на следующем шаге работы мастером пользователь может задать описании подписываемого файла и (или) адрес в Интернете с этой информацией.

На предпоследнем шаге диалога с мастером можно указать на необходимость добавления к ЭЦП отметки времени (в этом случае потребуется задать адрес Web-сервера службы времени).

На последнем шаге взаимодействия с мастером пользователю для подтверждения выводятся сведения о характеристиках подписываемого файла и ЭЦП.

Если вычисление ЭЦП завершилось успешно, то выводится соответствующее сообщение.

Для файлов, снабженных ЭЦП, в окне свойств, отображаемом командой «Свойства» контекстного меню, появляется вкладка «Цифровые подписи». На этой вкладке будут отображаться сведения об имени создателя подписи, его адресе электронной почты и отметке времени. После выделения подписи в списке с помощью кнопки «Сведения» можно получить дополнительные сведения об ЭЦП и ее создателе.

Для проверки истинности подписанного ЭЦП файла может использоваться утилита SignTool с командой verify. Синтаксис командной строки при вызове этой утилиты следующий:

SignTool verify [*опции*] *имя подписанного файла*

Для утилиты SignTool с командой verify можно указать до четырех опций: /q (при успешном завершении не генерируется никаких сообщений, а при ошибке – минимальное количество сообщений), /v (отображение полной информации об истинности подписанного файла), /r *имя* (имя владельца корневого сертификата, удостоверяющего сертификат подписи) и /tw

(генерируется предупреждение, если подписанный файл не имеет отметки времени). Утилита SignTool с командой verify определяет тип проверяемого подписанного файла автоматически.

Если подпись корректна, то в командной строке выводится соответствующее сообщение, содержащее имя подписанного файла и результат его проверки, например:

```
SignTool verify my.stl
Successfully verified: my.stl
```

Если проверка подписи завершилась неудачно, то выводится сообщение о причинах ошибки, например:

```
SignTool Error: A certificate chain processed, but terminated in a root
    certificate which is not trusted by the trust provider.
SignTool Error: File not valid: my.stl
```

Если при вызове утилиты signtool с командой verify была указана опция /v, то выводится более полная информация, например:

```
signtool verify /v test.exe
Verifying: test.EXE
SignTool Error: WinVerifyTrust returned error: 0x800B010D
    Signing Certificate Chain:
  Issued to: Root Agency
  Issued by: Root Agency
  Expires:   01.01.2040 3:59:59
  SHA1 hash: FEE449EE0E3965A5246F000E87FDE2A065FD89D4
    Issued to: my cert
    Issued by: Root Agency
    Expires:   01.01.2040 3:59:59
    SHA1 hash: 5F88F525BEF3DD95A54BFFDE5B607132978648D7
File is not timestamped.
SignTool Error: File not valid: test.EXE
Number of files successfully Verified: 0
Number of warnings: 0
Number of errors: 1
```

В этом примере содержится информация обо всех сертификатах построенной при проверке ЭЦП цепочки сертификатов и указывается на отсутствие отметки времени в проверяемом файле.

4. Управление сертификатами

Утилита CertMgr обеспечивает поддержку управления сертификатами, списками доверенных сертификатов (CTL) и списками отозванных сертификатов (CRL). К основным функциям этой системной программы относятся:

- отображение информации из сертификатов, CTL и CRL;
- копирование сертификатов, CTL и CRL из одного хранилища сертификатов в другое;

- удаление сертификатов, CTL и CRL из хранилища;
- экспорт (сохранение) закодированных сертификатов, CTL и CRL из хранилища в файл;
- импорт (загрузка) закодированных сертификатов, CTL и CRL из файла в хранилище сертификатов.

Формат командной строки при вызове утилиты CertMgr следующий:
CertMgr [/add | /del | /put][*опции*] [/s [/r *раздел реестра*]]
[*входное имя*] [/s [/r *раздел реестра*]] [*выходное имя*]

В табл. 7 приведено описание флагов операций, выполняемых утилитой CertMgr.

Табл. 7

Флаг операции	**Описание**
не задан	Отображение сертификатов, CTL и CRL
/add	Копирование сертификатов, CTL и CRL в хранилище сертификатов
/del	Удаление сертификатов, CTL и CRL из хранилища
/put	Экспорт сертификатов, CTL и CRL из хранилища в файл

Если флаг операции не задан, то отображаются все сертификаты, CTL и CRL из файла с сохраненным хранилищем или самого хранилища сертификатов, чье имя задается в качестве входного имени (выходное имя в этом случае не используется).

Если задан флаг операции /add, то входное имя – это имя хранилища сертификатов, содержащее сертификаты, CTL и CRL, которые будут добавлены в хранилище, чье имя задано как выходное имя. В качестве выходного имени может быть задано имя файла с сохраненным хранилищем. Если задана опция /7, то хранилище сохраняется в файле формата PKCS #7 (опция /7 не может применяться, если выходное имя указывает на системное хранилище сертификатов).

Если задан флаг операции /del, то входное имя определяет имя хранилища сертификатов, CTL и CRL, а выходное имя – имя хранилища, в которое будут помещены копии элементов входного хранилища, оставшихся в нем после удаления. Если выходное имя не задано, то модифицируется входное хранилище. В качестве выходного имени может быть указано имя файла с сохраненным хранилищем или (при задании опции /7) в формате PKCS #7. Опция /7 не применяется, если выходное имя указывает на системное хранилище сертификатов.

Если задан флаг операции /put, то входное имя – это имя хранилища сертификатов, закодированные элементы которого записываются в файл в формате X.509, задаваемый выходным именем (при указании опции /7 выходной файл имеет формат PKCS #7).

В табл. 8 приведено описание опций, которые могут быть указаны при вызове утилиты CertMgr.

Если пользователю требуется обработать несколько элементов хранилища в одной из трех возможных категорий, то можно использовать три возможности:

- указать опцию /all, чтобы копировать, сохранить или удалить все элементы указанной категории;
- задать опции /n и (или) /sha1 для точного указания на нужные элементы;
- если не заданы опции /all, /n и /sha1, то утилита CertMgr отображает пользователю список элементов хранилища для обработки, а пользователь должен указать индекс нужного ему элемента.

Помимо системных хранилищ сертификатов утилита CertMgr может работать с файлами хранилищ следующих форматов:

- закодированных сертификатов, CTL или CRL (должна использоваться кодировка CRYPT_STRING_BASE64);
- PKCS #7;
- подписанного сообщения;
- ранее сохраненного хранилища сертификатов.

Тип файла с хранилищем сертификатов может не указываться, т.к. утилита CertMgr может сделать это сама и выполнить необходимое действие.

Тип провайдера хранилища сертификатов также может не указываться, т.к. утилита CertMgr выберет нужный тип в зависимости от типа хранилища или файла.

Табл. 8

Опция	**Флаг операции**	**Описание опции**
/v	не задан	Отображение полной информации об элементах хранилища
/c	любой	Обрабатываются только сертификаты
/CTL	любой	Обрабатываются только CTL
/CRL	любой	Обрабатываются только CRL
/all	/add /del /put	Обрабатываются все элементы заданного типа
/e *тип*	любой	Тип кодировки сертификата (X509_ASN_ENCODING по умолчанию)
/y *имя*	любой	Имя провайдера хранилища сертификатов
/7	/add /del /put	Сохранение в файле формата PKCS #7
/f *флаги*	любой	Флаги открытия хранилища сертификатов (по умолчанию 1, т.е. хранилище открывается в профиле пользователя, более полная информация содержится в описании функции CertEnumSystemStore); эта опция может использоваться только при задании опции /y
/n *имя*	/add /del /put	Часть имени сертификата (используется только для элементов-сертификатов)
/sha1 *хеш*	/add /del /put	Хеш-значение сертификата, CTL или CRL

/s	любой	Указание на то, что хранилище сертификатов является системным
/r *раздел*	любой	Размещение системного хранилища сертификатов в реестре: currentUser (используется раздел HKEY_CURRENT_USER, это значение по умолчанию) или localMachine (HKEY_LOCAL_MACHINE); эта опция используется только вместе с /s

Рассмотрим примеры вызова утилиты CertMgr из командной строки. В первом примере утилита вызывается для просмотра полной информации о списках доверенных сертификатов, содержащихся в системном хранилище My:

```
CertMgr /v /CTL /s My
==============CTL # 1 ==========
SubjectUsage::
  [0] 1.3.6.1.5.5.7.3.3
ListIdentifier::
      4D 00 79 00 00 00                         'M.y...'
ThisUpdate::
  Sat Mar 11 10:26:00 2006
NextUpdate::
  Not Available
SHA1 Thumbprint::
     6571454F ED8E3E88 5B89624C 3489E6C2 48EC6333
MD5 Thumbprint::
     B8AC2CA5 DA2DE58F 611B8A8C D91391B7
Version:: 0
SubjectAlgorithm:: 1.3.14.3.2.26
SubjectAlgorithm.Parameters::
    05 00                                     '..'
-----  Entries  -----
 [0] SubjectIdentifier::
      4F BC 39 A3 B3 D9 03 0B 19 8C C6 33 CA 81 8B 08    'O.9.......3....'
      8F 86 DD EE                                 '....'
No signer
================================================
CertMgr Succeeded
```

Отображаемая информация содержит идентификатор объекта для назначения CTL, понятное имя CTL (в кодировке Unicode), дату и время создания списка и другие сведения.

В следующем примере просматриваются все элементы каждой из трех возможных категорий, содержащиеся в хранилище в файле myFile.ext:

```
CertMgr myFile.ext
```

В третьем примере просматриваются все элементы хранилища My:

```
CertMgr /s my
```

Если элементов каких-то категорий нет, то выводится соответствующее сообщение. Например, при отсутствии списков отозванных сертификатов выводится строка:

==========No CRLs ==========

В следующем примере все сертификаты, хранящиеся в файле myFile.ext, копируются в файл newFile.ext:

CertMgr /add /all /c myFile.ext newFile.ext

В пятом примере сертификат, чье имя содержит myCert, копируется из системного хранилища My в файл newCert.cer:

CertMgr /add /c /n myCert /s my newCert.cer

В следующем примере удаляются все сертификаты из системного хранилища My:

CertMgr /del /all /c /s MY

В седьмом примере все списки доверенных сертификатов их хранилища My удаляются, а результирующее хранилище помещается в файл newStore.str:

CertMgr /del /all /ctl /s my newStore.str

В следующем примере вначале создается самоподписанный сертификат, помещаемый в файл sign.cer (секретный ключ помещается в файл sign.pvk):

MakeCert /sv sign.pvk /r /n "CN=THIS IS A TEST OF MAKECTL" sign.cer

Затем созданный самоподписанный сертификат преобразуется в сертификат SPC, после чего pvk и spc-файлы преобразуются в pfx-файл:

Cert2SPC sign.cer sign.spc

Pvk2pfx /pvk sign.pvk /spc sign.spc /pfx sign.pfx

Далее создается другой самоподписанный сертификат, который помещается в файл test.cer:

MakeCert /sv test.pvk /r /n "CN=THIS IS MY TEST CERT" test.cer

Второй созданный сертификат также преобразуется в сертификат SPC и pfx-файл:

Cert2SPC test.cer test.spc

Pvk2pfx /pvk test.pvk /spc test.spc /pfx test.pfx

Затем сертификат из файла test.cer включается в новый список доверенных сертификатов, который сохраняется в файле test.stl:

MakeCTL test.cer test.stl

Созданный CTL подписывается с помощью ранее созданного файла sign.pfx:

SignTool sign /a /f sign.pfx test.stl

Подписанный CTL перемещается в системное хранилище доверенных сертификатов trust:

CertMgr /add /ctl test.stl /s trust

Сертификат из файла sign.cer помещается в системное хранилище корневых доверенных сертификатов root:

CertMgr /add /c sign.cer /s root

Файл test.exe подписывается с помощью секретного ключа и сертификата из файла test.pfx:

SignTool sign /a /f test.pfx test.exe

И, наконец, проверяется аутентичность файла test.exe:

SignTool verify test.exe

В заключительном примере сертификат, содержащий в имени подстроку myCert и находящийся в системном хранилище сертификатов Root, сохраняется в закодированном виде в файле newCert.cer:

certmgr /put /c /n myCert /s root newCert.cer

Если утилита CertMgr вызывается без опций операционной системе Windows NT с SP4, Windows 2000, Windows ME и старше, то начинается взаимодействие с менеджером управления сертификатами. С помощью вкладок этого диалогового окна можно управлять сертификатами из системных хранилищ личных сертификатов, сертификатов других пользователей, промежуточных центров сертификации (удостоверяющих центров) и доверенных корневых центров сертификации.

С помощью элемента «Дополнительные назначения» списка «Назначение» и кнопки «Дополнительно» можно отбирать сертификаты с требуемым назначением.

С помощью кнопки «Просмотр» (после выделения нужного сертификата в списке) можно ознакомиться с характеристиками сертификата. На вкладке «Общие» выводится информация о назначении, владельце и издателе сертификата, его сроке действия и существовании связанного с сертификатом секретного (личного) ключа.

На вкладке «Состав» отображается полная информация о сертификате. Кнопка «Копировать в файл» позволяет начать диалог с мастером экспорта сертификатов, а кнопка «Свойства» - отменить некоторые из имеющихся назначений сертификата.

Вкладка «Путь сертификации» позволяет просмотреть список сертификатов, образующих цепочку сертификации выбранного сертификата, а также результат ее проверки.

Кнопка «Импорт» на вкладке «Общие» менеджера управления сертификатами позволяет начать диалог с мастером импорта сертификатов.

На втором шаге работы с мастером пользователь должен выбрать файл, содержащий импортируемый сертификат. Возможен выбор файла одного из следующих типов:

- файла формата обмена персональной информацией PFX, определенного в стандарте PKCS #12;
- файла с закодированным криптографическим сообщением, формат которого определен в стандарте PKCS #7;
- файла с ранее сохраненным хранилищем сертификатов;
- файла с закодированным сертификатом, формат которого определен в стандарте X.509;
- файла с закодированным списком доверенных сертификатов;

- файла с закодированным списком отозванных сертификатов.

На третьем шаге работы с мастером требуется указать хранилище для импортируемого сертификата. При выборе переключателя «Поместить все сертификаты в следующее хранилище» можно с помощью кнопки «Обзор» явно указать требуемое хранилище сертификатов. Выключатель «Показать физические хранилища» позволяет выбрать конкретное размещение выбираемого хранилища.

При выборе переключателя «Автоматически выбрать хранилище на основе типа сертификата» выбор хранилища для импортируемого сертификата будет произведен мастером.

На заключительном шаге взаимодействия с мастером импорта сертификатов пользователю для подтверждения отображаются сведения о файле с импортируемым сертификатом и хранилище, куда он будет помещен.

При успешном завершении импорта выбранного сертификата отображается окно с соответствующим сообщением.

При нажатии на кнопку «Экспорт» (после выбора нужного сертификата в списке) на вкладке «Общие» окна менеджера сертификатов можно инициировать диалог с мастером экспорта сертификатов.

На втором шаге работы с этим мастером пользователь должен выбрать форму экспорта сертификата – вместе с соответствующим ему секретным ключом или без него.

В зависимости от сделанного на втором шаге выбора пользователю на третьем шаге работы с мастером будет предложено выбрать формат файла при экспорте сертификата.

При сохранении сертификата вместе с секретным ключом можно выбрать только формат PKCS #12 с дополнительными возможностями:

- включением, если возможно, всех сертификатов в цепочке сертификации;
- включением усиленной защиты секретного ключа;
- удалением секретного ключа после его успешного экспорта в файл.

При экспорте в файл только сертификата (без соответствующего секретного ключа) возможен выбор одного из следующих форматов:

- в закодированном двоичном виде в соответствии со стандартом X.509;
- в закодированном (с использованием способа кодировки CRYPT_STRING_BASE64) виде в соответствии со стандартом X.509;
- в виде закодированного криптографического сообщения в соответствии со стандартом PKCS #7 (с возможным включением всех сертификатов в цепочке сертификации).

Если экспортируется сертификат вместе с секретным ключом, то на следующем шаге работы с мастером пользователю предлагается задать пароль для генерации сеансового ключа шифрования экспортируемого секретного ключа.

На следующем шаге взаимодействия с мастером экспорта сертификатов требуется задать имя файла для экспорта сертификатов.

Расширение имени введенного имени файла будет выбрано мастером автоматически на основе ранее указанных пользователем характеристик.

На последнем шаге работы с мастером пользователь должен подтвердить ранее введенные данные для экспорта выбранного им сертификата. Если экспорт сертификата завершится успешно, то будет выведено соответствующее сообщение, иначе появится сообщение об ошибке.

При выборе сертификата в списке на вкладке «Общие» окна менеджера сертификатов становится доступной кнопка «Удалить». Нажатие на нее приведет к появлению диалогового окна с запросом подтверждения удаления сертификата.

При использовании обозревателя Internet Explorer версии 5.0 и старше возможен вызов менеджера сертификатов с помощью кнопки «Сертификатов…» на вкладке «Содержание» окна свойств обозревателя (это окно доступно также с помощью Панели управления Windows).

Если при проверке ЭЦП с помощью утилиты SignTool с командой verify пользователь с помощью выключателя «Всегда доверять содержимому, полученному от…» внес имя автора подписи в список доверенных издателей, то с помощью кнопки «Издателей…» на вкладке «Содержание» окна свойств обозревателя можно просмотреть список доверенных издателей и, при необходимости, удалить из него отдельные сертификаты.

5. Получение сертификата в удостоверяющем центре

В операционных системах Microsoft Windows 2000/XP/2003 для управления сертификатами может быть использована *оснастка* (snap-in) «Сертификаты». Эта системная программа может быть добавлена в *консоль управления Microsoft* (Microsoft Management Console, MMC), которая может быть вызвана с помощью команды Пуск | Выполнить | mmc. Для добавления оснастки «Сертификаты» требуется использовать команду меню Консоль | Добавить / удалить оснастку. В появившемся диалоговом окне нужно нажать кнопку «Добавить» и в списке доступных оснасток выбрать «Сертификаты».

С помощью оснастки «Сертификаты» можно просматривать содержимое хранилищ сертификатов, просматривать, импортировать и экспортировать сертификаты (аналогично менеджеру сертификатов, рассмотренному в предыдущем разделе). Эти операции доступны с помощью команд меню оснастки и с помощью контекстного меню соответствующих объектов (хранилищ и сертификатов). Дополнительно доступна команда «Поиск сертификата» в одном или нескольких хранилищах, позволяющая искать нужные сертификаты по именам издателя или владельца, серийному номеру, хеш-значению и другим критериям.

К другим дополнительным возможностям оснастки «Сертификаты» относятся просмотр списков отозванных сертификатов и запрос сертификата в удостоверяющем центре. Для просмотра списка отозванных сертификатов (CRL) нужно выделить узел «Список отзыва сертификатов» в левой части

окна оснастки и двойным щелчком на имени списка в правой части этого окна открыть окно свойств CRL.

На вкладке «Общие» окна свойств CRL содержатся свойства самого списка (номер версии, название поставщика-издателя, сроки начала действия и следующего обновления и т.п. На вкладке «Список отзыва» содержатся сведения о входящих в CRL отозванных сертификатов (серийных номерах, датах отзыва и т.п.).

Запрос и получение сертификата в оснастке «Сертификаты» возможно с помощью команды «Запросить новый сертификат» контекстного меню хранилища сертификатов «Личные», которая начинает диалог пользователя с мастером запроса сертификатов. Это становится возможным, если в локальной сети установлен удостоверяющий центр организации (enterprise certificate authority), иначе будет выведено сообщение об ошибке.

На первом шаге работы с мастером необходимо выбрать шаблон для запрашиваемого сертификата. Если на этом шаге включить выключатель «Дополнительные параметры», то у пользователя появится возможность выбора криптопровайдера, который обеспечит генерацию открытого и секретного ключей для запрашиваемого сертификата. При этом можно включить параметр усиленной защиты секретного ключа, при которой пользователь будет уведомляться при любой попытке приложения использовать его секретный ключ.

Начиная с версии Windows XP, на этом дополнительном шаге работы с мастером запроса сертификатов можно также выбрать длину генерируемого ключа асимметричного шифрования и указать на возможность впоследствии экспортировать секретный ключ, связанный с запрашиваемым сертификатом.

Вторым дополнительным шагом работы с мастером (при включенном флажке «Дополнительные параметры») является возможность задания пользователем удостоверяющего центра (центра сертификации), в который будет отправлен запрос. С помощью кнопки «Обзор» можно отобразить окно для выбора одного из доступных удостоверяющих центров.

На предпоследнем шаге работы с мастером запроса сертификатов пользователю потребуется ввести понятное имя и описание сертификата. Эта информация может потребоваться при поиске нужного сертификата.

На заключительном шаге работы с мастером пользователю для подтверждения выводится информация о сформированном запросе на получение сертификата. При успешной отправке запроса и выдаче сертификата выводится диалоговое окно с подтверждающим сообщением, в котором также есть кнопки для просмотра полученного сертификата и его установки в системе. При нажатии на кнопку «Установить сертификат» и успешном завершении этой процедуры выводится соответствующее сообщение.

Выданный сертификат помещается в хранилище для личных сертификатов пользователя.

Независимо от того, какой удостоверяющий центр установлен в компьютерной системе – организации (enterprise) или автономный

(изолированный, standalone), пользователи могут подавать запросы на изготовление сертификатов через Web-интерфейс. Это будет возможно, если на компьютере с удостоверяющим центром будет также установлена служба IIS (Internet Information Services) с поддержкой активных серверных страниц (active server pages, ASP).

Для начала процедуры подачи запроса на изготовление сертификата через Web-интерфейс пользователь должен в адресной строке обозревателя Internet Explorer ввести http://*сетевое имя компьютера с удостоверяющим центром*/certsrv. Если установленная на указанном компьютере служба сертификатов поддерживает запросы сертификатов через Web-интерфейс, то пользователь увидит начальную страницу.

На начальной странице пользователь может выбрать одно из трех действий:

- запросить сертификат;
- просмотреть состояние ранее поданных запросов на сертификаты;
- загрузить на свой компьютер сертификат удостоверяющего центра (центра сертификации, ЦС), цепочку сертификации для проверки своего сертификата и список отозванных сертификатов;
- получить документацию служб сертификатов Microsoft Windows с Web-узла этой корпорации.

При выборе запроса сертификата пользователь в следующем окне должен будет выбрать тип требующегося ему сертификата (например, Web-обозревателя или защиты электронной почты) или выдать расширенный запрос сертификата, при котором он сможет задать индивидуальные параметры этого запроса.

При выборе расширенного запроса сертификата пользователь затем сможет либо создать и выдать оригинальный запрос к удостоверяющему центру, либо выдать запрос на основе уже имеющегося файла в формате стандартов PKCS #10 или PKCS #7.

Для создания и выдачи оригинального запроса сертификата на компьютер пользователя потребуется загрузить и установить необходимые элементы управления ActiveX. Если политикой безопасности для компьютера пользователя это запрещено, то будет выдано соответствующее сообщение и создать запрос будет невозможно.

В окне расширенного запроса сертификата пользователь сможет:

- ввести идентифицирующие себя сведения (имя, адрес электронной почты, организацию, подразделение, город, область и двухсимвольный код страны);
- выбрать из списка нужный тип сертификата (например, проверки подлинности клиента, защиты электронной почты, подписи кода и т.п.);
- задать параметры создаваемого ключа асимметричного шифрования:
    - ♦ выбрать один из переключателей – создать новый или использовать существующий набор ключей;

- ♦ выбрать из списка криптопровайдеров, установленных на его компьютере, тот CSP, который будет использован для создания и хранения пары асимметричных ключей;
- ♦ с помощью одного из переключателей выбрать использование создаваемых ключей – Exchange (обмен сеансовыми ключами при симметричном шифровании), подпись или оба;
- ♦ ввести размер создаваемого асимметричного ключа, используя выводимую рядом с редактируемой строкой информацию о минимальном, максимальном и стандартных размерах ключей, поддерживаемых выбранным криптопровайдером;
- ♦ с помощью одного из переключателей выбрать вариант с использованием автоматически назначаемого или заданного пользователем имени контейнера ключей;
- ♦ пометить создаваемый секретный ключ как экспортируемый, что позволит в дальнейшем экспортировать запрашиваемый сертификат вместе с секретным ключом (возможен и экспорт ключа в файл одновременно с созданием запроса на получение сертификата);
- ♦ включить режим усиленной защиты секретного (закрытого) ключа, что приведет к выдаче запроса на подтверждение любого использования секретного ключа пользователя в приложении;
- ♦ включить режим сохранения сертификата в локальном хранилище сертификатов (т.е. в разделе реестра HKEY_LOCAL_MACHINE), что потребует наличие привилегий администратора (по умолчанию сертификат будет сохранен в хранилище, размещаемом в разделе реестра HKEY_CURRENT_USER);

- задать дополнительные параметры:
  - ♦ выбрать формат запроса – CMC (Common Messaging Calls, кроссплатформенного набора функций, обеспечивающего независимость от используемых систем передачи сообщений, операционных систем, используемый аппаратных средств) или стандарта PKCS #10;
  - ♦ выбрать в списке алгоритмов хеширования, поддерживаемых выбранным криптопровайдером, требуемый алгоритм хеширования запроса сертификата;
  - ♦ включить режим сохранения запроса в файле, путь к которому нужно будет задать здесь же, что приведет к тому, что созданный запрос сертификата не будет немедленно направлен удостоверяющему центру;
  - ♦ ввести понятное имя сертификата для облегчения его дальнейшего использования.

После ввода всех необходимых пользователю параметров расширенного запроса сертификата необходимо нажать на кнопку «Выдать».

Для подтверждения отправки запроса сертификата будет выведено соответствующее сообщение, после чего, непосредственно перед передачей запроса, предупреждение обозревателя Internet Explorer о возможном перехвате информации из запроса. Если пользователь подтвердил необходимость отправки запроса, то он направляется удостоверяющему центру, обрабатывается им и, в случае успеха, сертификат выдается пользователю, о чем он извещается в специальном окне обозревателя Internet Explore.

В окне подтверждения выдачи сертификата располагается гиперссылка «Установить сертификат», с помощью которой пользователь сможет поместить выданный ему сертификат в хранилище личных сертификатов на своем компьютере. При этом будет отображено окно подтверждения установки сертификата на компьютере пользователя. При подтверждении пользователем установки сертификата ему будет выведено предупреждение о возможности доступа других лиц к передаваемому по сети сертификату.

После подтверждения пользователем передачи сертификата он пересылается на компьютер пользователя и сохраняется в хранилище личных сертификатов, после чего пользователю отображается окно с сообщением об успешной установке сертификата.

Если пользователь имеет файл с уже подготовленным запросом на выдачу сертификата, он может создать запрос на основе этого файла для выдачи нового (в формате PKCS #10) или обновления уже существующего (в формате PKCS #7) сертификата. Для этого в соответствующем окне он должен в поле «Сохраненный запрос» ввести или выбрать с помощью соответствующей ссылки имя файла с запросом, задать при необходимости дополнительные атрибуты для запрашиваемого сертификата и нажать кнопку «Выдать».

При создании запроса сертификата на основе уже существующего шаблона (типа) пользователь может ввести идентифицирующие сведения и задать дополнительные параметры:

- выбрать имя поставщика службы криптографии (криптопровайдера, CSP) в списке установленных на его компьютере CSP;
- включить режим усиленной защиты секретного ключа;
- выбрать формат запроса:
- при желании воспользоваться формой расширенного запроса.

При выборе функции загрузки сертификата ЦС, цепочки сертификатов ЦС и последнего базового CRL пользователю будет отображено окно для дальнейшего выбора требуемой ему подфункции. В списке «Сертификат ЦС» пользователь может выбрать нужный ему сертификат удостоверяющего центра. С помощью переключателей DER (Distinguished Encoding Rules, правила отличительного кодирования двоичных последовательностей) и Base 64 можно выбрать требуемый пользователю метод кодирования запрашиваемой информации.

Чтобы доверять сертификатам, выданным удостоверяющим центром, пользователь должен установить цепочку сертификатов ЦС. При выборе этой

ссылки будет отображено окно для подтверждения необходимости установки сертификатов, после чего (в случае положительного ответа) цепочка сертификации будет передана на компьютер пользователя, а входящие в нее сертификаты помещены в соответствующее хранилище, о чем пользователю будет выведено соответствующее сообщение в новом окне обозревателя Internet Explorer.

При выборе пользователем функций загрузки на свой компьютер сертификата ЦС, цепочки сертификатов ЦС и последнего базового списка отозванных сертификатов (CRL) ему будут последовательно отображены следующие окна:

- о начале загрузки с возможностью открытия файла с запрошенной информацией, сохранением его или отменой загрузки;
- стандартное диалоговое окно Windows «Сохранить как» для выбора папки и, при желании, имени сохраняемого файла;
- о завершении загрузки с возможностью открыть полученный сертификат удостоверяющего центра, цепочку сертификатов удостоверяющего центра или список отозванных сертификатов.