ЗАО «С-Терра СиЭсПи»
124460, г. Москва, Зеленоград, проезд 4806, д.6, этаж 4-й
Телефон: +7 (499) 940 9061
Факс: +7 (499) 940 9061
Эл.почта: information@s-terra.com
Сайт: http://www.s-terra.com

Программный комплекс ”Шлюз безопасности CSP VPN Gate. Версия 3.1”

Руководство администратора

Приложение

РЛКЕ.00005-01 90 03

24.05.2011

Содержание

Приложение

Описание формата ini-файлов

Формат допустимых строк

1. Все `ini-`файлы, используемые в продукте CSP VPN Gate, могут содержать следующие типы строк:
   - *Строка имени секции* – строка, обозначающая начало новой секции переменных
   - *Строка описания переменной* – строка, содержащая имя и значение переменной
   - *Строка комментариев* – строка, содержащая комментарии или пустая строка.

Строка имени секции имеет следующий формат:

```
[SECTION_NAME]
```

Пробелы и символы табуляции, стоящие до открывающей и после закрывающей квадратных скобок, игнорируются. Именем секции считается строка, помещенная между квадратными скобками (любой символ является значимым). Допускается пустое имя секции.

```
[]
```

Строка описания переменной имеет следующий формат:

```
var_name = var_value
```

Пробелы и знаки табуляции, стоящие до и после `var_name` и `var_value` игнорируются.

Строка комментариев имеет следующий формат:

```
! comment string
```

Пробелы и знаки табуляции, стоящие до символа `!` игнорируются.

Пустая строка (не содержащая ничего, кроме пробелов и знаков табуляции) приравнивается к строке комментария.

Любая строка, не удовлетворяющая ни одному типу описанных строк, является ошибочной, и будет приводить к ошибке чтения `ini-файла`.

2. Повторяющиеся имена секций.

В файле допустимо многократное использование *строки имени секции* с одним и тем же именем секции. В этом случае каждая последующая секция считается продолжением предыдущих (*строки описания переменных* объединяются)

3. Повторяющиеся имена переменных.

В файле допустимо многократное использование *строки описания переменной* с одним и тем же именем переменной. В том случае, если эти строки принадлежат к одной секции – действительным считается значение, описанное последней строчкой (*строки описания переменных* накладываются).

4. Переменные, не принадлежащие ни одной из секций.

В файле допускается указание *строк описания переменных*, не принадлежащих ни одной секции (в начале файла идут строки переменных без задания имени секции). Этот случай эквивалентен заданию секции с пустым именем. Следующие ситуации эквивалентны:

```
File01.ini
 Var01=value01
 Var02=value02
 Vat03=valeue03


File02.ini
 []
 Var01=value01
 Var02=value02
 Vat03=valeue03
```

5. Перезапись ini-файла.

В процессе работы некоторые `ini`-файлы могут модернизироваться продуктом. При этом, все комментарии и пустые строки будут сохранены.

- В случае повторяющихся имен секций подобные секции будут объединены в одну (первая дополняется переменными последующих). В этом случае комментарии, принадлежащие секциям, объединяются.
- В случае повторяющихся имен переменных (принадлежащих одной секции) будет оставлено только последнее ее описание. В этом случае комментарии, принадлежащие переменной, объединяются.

6. Принадлежность строк комментариев.

Любая *строка комментариев*, расположенная перед *строкой имени секции* или *строкой описания переменной*, считается принадлежащей этой строке.

Конвертор

В данной главе описана внутренняя логика работы конвертора из Cisco-like конфигурации в Native-конфигурацию, управление конвертированием с помощью INI-файла, формирование имен структур при конвертировании, а также указан список сообщений, предупреждений и ошибок, которые могут появиться при протоколировании событий.

Основная логика работы

1. Конвертор выполнен в виде динамической библиотеки s_converter.dll (Win32) / libs_converter.so (Solaris). Также используются некоторые вспомогательные файлы и агентские библиотеки.
2. Конвертор работает в рамках программы cs_console.
3. При выходе из конфигурационного режима, если в конфигурацию были внесены какие-то изменения, cs_console вызывает конвертор и передает ему внутреннее представление Cisco-конфигурации.
4. Во время работы конвертора используются настройки конвертора, описанные в разделе "Внутренние настройки консоли и конвертора". Некоторые из настроек могут редактироваться пользователем. В результатае работы конвертора формируется LSP в Native-формате.
5. Логика формирования имен структур в Native-конфигурации представлена в разделе "Формирование имен структур LSP при конвертировании".
6. Далее происходит попытка загрузки LSP в Native-формате в агента.
   - Если по каким-то причинам произошла ошибка при загрузке, Native-конфигурация пишется в файл `erroneous_lsp.txt`, расположенный в:

     `Windows` – в каталоге агента

     `Solaris` и `Linux` – в каталоге `/var/cspvpn`.
7. В конце работы выдается результат (успех/неуспех) обратно в cs_console.

Ограничения на конвертор

1. Поддерживается набор команд, определенный в документе «Cisco-like команды»..
2. В Cisco используется примерно следующая логика работы с access list:

   `<interface_acl> -> <crypto_map_acl> -> <interface_acl>`,

   где `<interface_acl>` – access list в интерфейсе,

   а `<crypto_map_acl>` – access list в crypto map.

   В конверторе используется логика разворачивания access list-ов в сквозную модель правил. При этом возможны некоторые несоответствия и несовместимости (подробнее см."Описание обработки интерфейсов").

   - В правилах в access list в маске подсети допускается указание только непрерывной линейки из установленных битов в конце (т.е. 00…01…1, например 0.0.0.255 0.0.0.63 и т.п.). Не допускается разрывов в полях установленных и сброшенных битов (например, маски вида 0.255.0.255). В случае появления запрещенной маски, конвертация завершается с ошибкой [3.9].

3. Ряд ограничений на `ca trustpoint`:
   - enrollment игнорируется (только ручное задание сертификатов).
   - Читаются только CA-сертификаты, локальные сертификаты игнорируются.
     - Небольшое пояснение: в Cisco по команде `crypro ca certificate chain` показываются CA сертификаты и локальные сертификаты. Через эту команду все сертификаты можно посмотреть, удалить и ввести CA сертификаты. Однако, локальные сертификаты нельзя ввести таким образом (они будут неработоспособны без секретного ключа). В `cs_console` данная команда используется только для работы с CA сертификатами.
   - Под обозначением RSA-сертификатов (другие в Cisco не используются) могут использоваться RSA, ГОСТ и DSA-сертификаты.
   - В Cisco в пределах одного `trustpoint` могут вписываться только сертификаты из одной цепочки. В `cs_converter` допускаются любые CA сертификаты.
   - Задается строгое соответствие: RSA CA сертификат подписывает только RSA-сертификаты, ГОСТ CA сертификат подписывает только ГОСТ сертификаты, DSA CA сертификат подписывает только DSA-сертификаты.
   - Следует учитывать, что в конфигурации не задается точных критериев выбора локального сертификата (в терминах Native LSP задается USER_SPECIFIC_DATA). В связи с этим возможны ситуации, при которых не установится соединение, если присутствуют больше одного локального сертификата, подписанного разными CA.
     - Пример подобной ситуации: у партнера не прописана посылка Certificate Request, и партнер ожидает от агента конкретный сертификат (который действительно присутствует), но агент по своим критериям выбирает другой сертификат, который не подходит партнеру.
     - Как правило, таких проблем не возникает, если соблюдаются следующие условия:
       - У обоих партнеров прописана отсылка Certificate Request. По умолчанию конвертер именно так и делает. Cisco в большинстве случаев поступает также.
       - Не используется `Aggressive Mode` при работе с сертификатами (экзотический случай).
       - У партнера должны быть явно указаны CA-сертификаты, которыми может быть подписан локальный сертификат агента. В Native LSP агента – атрибут `AcceptCredentialFrom` (cs_converter вписывает все CA-сертификаты, лежащие в базе). В Cisco – должен быть прописан подходящий `trustpoint`.
4. Ограничение на LDAP url: допускается только задание IP-адреса и, возможно, порта. Если задано DNS-name – данный url игнорируется.
5. Допускается только одно ISAKMP правило для одного IPsec-правила.
6. Если для данного crypto-map удалось подобрать несколько ISAKMP policy с разными Transform и методами аутентификации, то формируется одна IKERule, в которой пишутся ВСЕ трансформы и методы аутентификации, что приводит к несколько иной логике (т.е. теряется связь между трансформами и методами аутентификации).

Пример

#Фрагмент исходной конфигурации:

```
crypto isakmp policy 1
encr des
hash md5
authentication rsa-sig
```

```
crypto isakmp policy 2
encr 3des
hash sha
authentication pre-share
group 2

crypto isakmp policy 3
encr aes 128
hash md5
authentication rsa-sig
```

#Фрагмент Native-LSP (в ситуации, когда подходят все три policy):

```
AuthMethodRSASign auth_ca(
...
)
AuthMethodPreshared IKE_auth_key_192_168_11_110(
...
)
IKERule IKE_router_mc_fastethernet0_0_crypto_1(
    Transform* = IKETransform(
        CipherAlg   *= "DES-CBC"
        HashAlg     *= "MD5"
        GroupID     *= MODP_768
        LifetimeSeconds = 86400
    ),
    IKETransform(
        CipherAlg   *= "DES3-K168-CBC"
        HashAlg     *= "SHA1"
        GroupID     *= MODP_1024
        LifetimeSeconds = 86400
    ),
    IKETransform(
        CipherAlg   *= "AES-K128-CBC-7"
        HashAlg     *= "MD5"
        GroupID     *= MODP_768
        LifetimeSeconds = 86400
    )
    MainModeAuthMethod  *= auth_ca, IKE_auth_key_192_168_11_110
    AggrModeAuthMethod  *= auth_ca, IKE_auth_key_192_168_11_110
    DoAutopass          = TRUE
)
```


7. В фильтрах, в которых прописан локальный адрес ANY, в Native-LSP прописывается диапазон 0.0.0.0..255.255.255.255.
8. Если в crypto map прописаны несколько peer, каждый из которых аутентифицируется по preshared key, то используется следующий подход:
   - прописывается туннель и аутентификация для первого по счету peer
   - для остальных peer-ов проверяются preshared keys:
     - если preshared key совпадает с ключом для первого peer, то этот peer прописывается в качестве туннеля;
     - если preshared key не совпадает с ключом для первого peer, то для данного peer формируется отдельный `AuthMethodPreshared, IKERule и IPsecAction`. При этом в `IKERule` прописывается параметр:

       ```
       IKEPeerIPFilter* = FilterEntry(
                          IPAddress *= <IP-адрес peer> )
       ```

       В этом случае в `FilteringRule` для данной `crypto map` перечисляется список сформированных `IPsecAction`.

Пример подобного случая

#Фрагмент исходной конфигурации:

```
crypto isakmp key 1234 address 1.1.1.1
crypto isakmp key 5678 address 2.2.2.2
...
crypto map cmap 1 ipsec-isakmp
set peer 1.1.1.1
set peer 2.2.2.2
...
```

#Фрагмент Native-LSP:

```
AuthMethodPreshared IKE_auth_cs_key_1_1_1_1 (
    RemoteID = IdentityEntry(
        IPv4Address *= 1.1.1.1
    )
    SharedIKESecret = "cs_key_1_1_1_1"
)
IKERule IKE_cmap_1 (
   IKEPeerIPFilter* = FilterEntry(IPAddress *= 1.1.1.1)
...
    AggrModeAuthMethod  *= IKE_auth_cs_key_1_1_1_1
    MainModeAuthMethod  *= IKE_auth_cs_key_1_1_1_1
...
)
IPsecAction cmap_1 (
    TunnelingParameters *= TunnelEntry(
        PeerIPAddress = 1.1.1.1
...
    )
    IKERule = IKE_cmap_1
```



```
)
AuthMethodPreshared IKE_auth_cs_key_2_2_2_2 (
    RemoteID = IdentityEntry(
        IPv4Address *= 2.2.2.2
    )
    SharedIKESecret = "cs_key_2_2_2_2"
)
IKERule IKE_cmap_1_1 (
    IKEPeerIPFilter* = FilterEntry( IPAddress *= 2.2.2.2 )
...
    AggrModeAuthMethod  *= IKE_auth_cs_key_2_2_2_2
    MainModeAuthMethod  *= IKE_auth_cs_key_2_2_2_2
...
)
IPsecAction cmap_1_1 (
    TunnelingParameters *= TunnelEntry(
        PeerIPAddress = 2.2.2.2
...
    )
...
    IKERule = IKE_cmap_1_1
)
FilteringRule Filter_...(
...
    Action *= ( cmap_1 ), ( cmap_1_1 )
)
```

Следует учитывать, что подобная конфигурация приведет к тому, что работа со вторым peer будет возможна только в качестве ответчика. В качестве инициатора работа возможна только с первым peer.

Рекомендуется по возможности избегать таких ситуаций. Для этого, в случае указания в crypto map нескольких peers, следует либо использовать аутентификацию по сертификатам либо, в случае использования аутентификации на preshared keys, использовать одинаковый ключ для всех peers, перечисленных в одной crypto map.

В подобной ситуации выдается сообщение [2.10].

Если присутствует подобная конфигурация с несовпадающими preshared ключами и, кроме того, существует аутентификация на сертификатах; тогда к вышеперечисленным наборам `AuthMethodPreshared`, `IKERule` и `IPsecAction` добавится еще один, описывающий аутентификацию на сертификатах. При этом в `IPsecAction` будут прописаны все peers.

#Пример фрагмента LSP (отличия от предыдущего примера):

```
IKERule IKE_cmap_1_2 (
...
    AggrModeAuthMethod  *= auth_ca
    MainModeAuthMethod  *= auth_ca
...
)
```

```
IPsecAction cmap_1_2 (
    TunnelingParameters *= TunnelEntry(
        PeerIPAddress = 1.1.1.1
...
    ),
    TunnelEntry(
        PeerIPAddress = 2.2.2.2
...
    )
...
    IKERule = IKE_cmap_1_2
)
FilteringRule Filter_... (
...
    Action *= ( cmap_1 ), ( cmap_1_1 ), ( cmap_1_2 )
)
```

9. Существует специфический подход в случае, если в `crypto map set` присутствует несколько `crypto maps`, а в их `crypto-map-acls` существуют пересечения по адресам, причем в части правил присутствует `permi`t, а в других правилах – `deny`. Подробнее логика конвертирования для данной ситуации описана в п.5 раздела "Описание обработки интерфейсов".
10. Возможен только симметричный маршрут, поэтому в интерфейсах используется только `access-group in`, без `access-group out`. Если `access-group out` задано, оно игнорируется с выдачей предупреждения.
11. Существуют особенности в настройке маршрутизации:

    Если добавить из консоли `routing`, который уже присутствует в системной таблице маршрутизации, то он будет добавлен в текущую конфигурацию с диагностикой в файле лога.

    При отгрузке сконвертированной конфигурации (по любой причине), из системной таблицы маршрутизации будут удалены все записи, добавленные из консоли, которые также могли существовавать и до запуска консоли (например, добавленные с помощью команды `route add`).
12. Существуют дополнительные команды, которые отсутствуют у Cisco:
    - команда `set pool` – задает IKE-CFG pool, привязанный к конкретной `crypto map`. Работает в конфигурационном режиме `crypto map` и `crypto dynamic-map`.

      особый случай – команда `set pool <none>` – убирает для конкретной `crypto map` или `crypto dynamic-map` настройки IKE-CFG, которые, возможно, выставлены с помощью команды `crypto map client configuration address` или `crypto dynamic-map client configuration address` (они работают для с`rypto map set`).
    - команда `crypto dynamic-map client configuration address`. Работает аналогично команде `crypto map client configuration address`, но для `dynamic map set`.
13. Команды для задания ограничений по трафику и времени имеют больший диапазон, чем в Cisco:

- `security-association lifetime kilobytes`: в Cisco `2560-536870912`, у нас – `1-4294967295`.
- `security-association lifetime seconds`: в Cisco `120-86400`, у нас – `1-4294967295`.
- `IKE lifetime (seconds)`: в Cisco 60-86400, у нас – `1-4294967295`.

Примечание: Cisco-like консоли как и в Cisco отсутствует возможность убрать ограничения по трафику и времени (unlimited).

14. Существуют особенности при настройке шлюза для работы с мобильным клиентом. Можно использовать один из двух подходов:
    - с точки зрения логики настройки CSP VPN Gate, в acl, привязанном к crypto dynamic map, в качестве remote-адресов для мобильных клиентов необходимо указывать any. Таким образом указывается, что допускается любой физический адрес мобильного клиента.
    - с точки зрения логики настройки Cisco, в acl, привязанном к crypto dynamic map, в качестве remote-адресов для мобильных клиентов указывается пул, из которого роутер раздает адреса мобильным клиентам. Таким образом указывается, что область действия этой crypto dynamic map распространяется только для мобильных клиентов из пула:

Пример

#Фрагмент конфигурации:

```
ip local pool p1 10.0.0.0 10.0.0.255
!
crypto ipsec transform-set ts1 esp-des esp-md5-hmac
 mode tunnel
!
ip access-list extended acl
 permit  ip  0.0.0.0 255.255.255.255 10.0.0.0 0.0.0.255
!
crypto dynamic-map   dmap 1
 match address acl
 set transform-set ts1
 set pool p1
!
crypto map cmap 1 ipsec-isakmp dynamic dmap
!
!
!
interface FastEthernet0/0
 ip address 10.0.15.100 255.255.0.0
 crypto map cmap
```

#Фрагмент сконвертированной LSP:

```
AddressPool p1
(
    IPAddresses *= 10.0.0.0..10.0.0.255
)
```

```
IPsecAction dmap_1
(
    TunnelingParameters *= TunnelEntry(
        DFHandling=COPY
    )
    ContainedProposals *= ( ESP_ts1 )
    IKERule = IKE_dmap_1
)

FilteringRule Filter_nil_acl_dmap_1
(
    LocalIPFilter *= FilterEntry(
               IPAddress *= 0.0.0.0..255.255.255.255 )
    PeerIPFilter  *= FilterEntry(
               IPAddress *= 10.0.0.0/24 )
    NetworkInterfaces *= "pcn0"
    Action *= ( dmap_1 )
)
```

в сконвертированной LSP видно, что в структуре IPsecAction dmap_1 в атрибуте TunnelEntry отсутствует поле PeerIPAddress, то в качестве IKE-партнера для шлюза может выступать мобильный клиент с любым физическим адресом

в тоже время, если мобильный клиент является пассивным IKE-CFG клиентом (IKECFGRequestAddress=FALSE – не инициирует посылку запроса на получение адреса из пула), то защищенное соединение построено не будет. Присылаемый мобильным клиентом его физический адрес в качестве identity QM не подходит под соответствующее FilteringRule и QM шлюзом отвергается

для успешного создания соединения при такой конфигурации шлюза мобильный клиент должен выступать в качестве активного IKE-CFG клиента (IKECFGRequestAddress=TRUE). В этом случае в качестве identity QM используется выданный клиенту адрес из пула и соответствующее FilteringRule на шлюзе срабатывает

15. Если задается dynamic map без указания set peer (обычная ситуация), то формируются цепочки правил для всех возможных вариантов аутентификации. Например, если заданы несколько preshared keys для разных хостов, то будут сформированы правила для всех этих preshared keys и соответствующих им хостов.

    если задается dynamic map с указанием set peer (экзотический, но допустимый вариант), то данная dynamic map конвертируется аналогично static map.

16. В TunnelingParameters прописывается значение DFHandling:

    используется значение crypto ipsec df-bit для интерфейса, если оно присутствует в конфигурации.

    в противном случае используется глобальное значение crypto ipsec df-bit.

Логика запуска конвертора

1. В базу локальных настроек добавляется признак источника загруженной LSP: из утилиты lsp_mgr или из cs_converter (в дальнейшем возможно расширение списка).
2. При входе в конфигурационный режим проверяется источник текущей LSP. Возможны следующие варианты:
   - LSP загружена из `cs_converter` (согласованные политики).
   - LSP загружена из другого источника или вообще не загружена (рассогласованные политики).
3. Кроме того, при старте проверяются следующие изменения:
   - добавление или удаление сертификата в базе локальных настроек
   - удаление preshared ключа, заданного в Cisco-like конфигурации
   - изменение состава сетевых интерфейсов в агенте (добавлены или удалены с помощью `if_mgr`)
   - изменение адреса сетевого интерфейса.

   Во всех этих случаях данные изменения могут отразиться на LSP, формируемой с помощью конвертора. Поэтому, если зафиксировано одно или несколько упомянутых изменений, то данная ситуация также трактуется как рассогласование политик. При этом проверка загруженной LSP уже не делается.

   Следует отметить, что проверка на данные ситуации делается только при старте консоли. При повторном входе в конфигурационный режим в рамках одной сессии считается, что данные изменения уже корректно отработаны. При этом проверка LSP делается при каждом входе в конфигурационный режим.
4. В файл `cs_conv.ini` добавляется новый параметр – режим синхронизации политик, который отвечает за логику работы консоли в случае второго варианта (рассогласование политик). См. описание настройки `policy_sync`.
   - Данный параметр влияет на конвертирование текущей конфигурации при входе в режим `configure terminal`.
   - Данный параметр влияет на включение и выключение инкрементальной конфигурации (для случая рассогласованной политики): если этот режим включен, то для некоторых команд (сейчас – `routing и SNMP traps`) формируется и загружается инкрементальная конфигурация немедленно после прописывания этой команды. Следует учитывать, что при выключенной инкрементальной настройке возможны побочные эффекты, связанные с командами редактирования маршрутизации (как минимум): возможно добавление неправильной команды с корректным синтаксисом (например, может быть добавлен маршрут через шлюз, к которому не определен маршрут). В этом случае команда добавится в Cisco-like конфигурацию, а затем при конвертировании этой конфигурации возникнет ошибка на стадии загрузки сконвертированной Native-LSP.
   - Подробнее логика работы данного параметра описана в таблице.
5. Логика запуска конвертора для разных вариантов:

Режим синхронизации политик	Включен (значение по умолчанию)	Выключен
Стартовая LSP загружена из cs_converter.	При входе в режиме `configure terminal` не происходит конвертирования. Инкрементальная настройка включена. Запуск конвертора осуществляется при выходе из режима `configure terminal` только в том случае, если были произведены какие-либо	

	изменения в Cisco-like конфигурации.	
Стартовая LSP загружена из другого источника или не загружена вообще. Выдается сообщение в лог [1.7].	При входе в конфигурационный режим делается попытка сконвертировать текущую Cisco-like конфигурацию. Далее логика различается в зависимости от того, удалось сконвертировать конфигурацию или нет: Если удалось сконвертировать: инкрементальная настройка включена предыдущая агентская конфигурация (если она есть) сохраняется в файл `non_cscons.lsp` (расположение файла см. в Примечании). Об этом выдается сообщение в лог [1.6]. Если по каким-либо причинам не удалось сохранить файл, то сообщается в лог [3.11]. запуск конвертора осуществляется при выходе из режима `configure terminal` только в том случае, если были произведены какие-либо изменения в Cisco-like конфигурации. Если не удалось сконвертировать, то выдается сообщение в лог [1.9]. Инкрементальное конфигу-рирование ВЫКЛЮЧЕНО, поскольку нет LSP, к которой можно было бы корректно приложить инкрементальную LSP (политики рассогласованы). При выходе из режима `configure terminal` вызывается конвертор только в том случае, если были произведены какие-либо изменения в Cisco-like конфигурации. Если изменений сделано не было, то данную конфигурацию не удастся сконвертировать, поэтому конвертор вызывать не нужно (сообщается в лог [2.8]).	При входе в режим `configure terminal` не происходит конвертирования. Инкрементальная настройка выключена. Выдается сообщение в лог [1.8]. При выходе из режима `configure terminal` вызывается конвертор вне зависимости от того, были внесены изменения в Cisco-like конфигурацию или нет. Если конвертирование прошло успешно, то предыдущая конфигурация агента (если она есть) сохраняется в файл `non_cscons.lsp` (расположение файла см. в Примечании). Об этом выдается сообщение в лог [1.6]. Если по каким-либо причинам не удалось сохранить файл, то сообщается в лог [3.11].

Примечание:

расположение файла `non_cscons.lsp` зависит от ОС:

`Windows` – в каталоге агента

`Solaris и Linux` – в каталоге `/var/cspvpn`.

6. Если при выходе из конфигурационного режима происходит конвертирование конфигурации, и это конвертирование завершается с ошибкой; то на консоль выдается сообщение об ошибке: `"LSP conversion failed. You can use the "show load-message" command to obtain the additional information."` ("Конвертирование

LSP завершилось с ошибкой. Вы можете использовать команду "`show load-message`" для получения дополнительной информации").

7. Для некоторых команд (сейчас – `routing` и `SNMP traps`) формируется и загружается инкрементальная конфигурация немедленно после прописывания этих команд, и для них существуют следующие особенности:
   - после обработки команды (неважно успешном или нет) запоминаются сообщения от cs_converter, которые можно посмотреть по команде do show load-message:
     - возможна ситуация, при которой команда выполнена успешно и на консоль не выдано никаких сообщений, но по команде do show load-message могут быть выданы информационные сообщения от конвертора
   - при ошибке конвертирования на консоль выдается сообщение:

     % LSP conversion failed. Reason:

     <Message_from_cs_converter>

     где <Message_from_cs_converter> – сообщение, полученное от cs_converter (аналогичное тому, что будет выдано по команде do show load-message).

     - если от cs_converter не получено сообщений, но на консоль выдается:

       % LSP conversion failed.

     Примечание: появление данного сообщения можно считать ошибкой в продукте – отсутствие корректной нотификации в cs_converter. Обратитесь в службу поддержки по адресу: support@s-terra.com.

Алгоритм работы конвертора

1. Подготовительный этап:

   Инициализация лога.

   Инициализация локальных настроек.

2. Написание служебной информации в комментариях:

   Фраза "`This is automatically generated LSP`".

   Дата и время конвертации.

3. Создание заголовка конфигурации:

   Служебная информация (версия и т.п.).

   Настройки LDAP и CRL-processing.

   Глобальные настройки лога.

4. Обработка интерфейсов. Подробнее см. "Описание обработки интерфейсов". При этом формируются правила фильтрации, в том числе и IPsec (APPLY).

5. Для APPLY правила происходит поиск подходящего ISAKMP правила:

   Сначала делается попытка найти подходящее правило на `Preshared key`.

   Также берется первое по счету правило `rsa-sig`.

6. Если подобрано правило на `Preshared key` – прописывается аутентификационная информация с соответствующим именем ключа.

   Для `main mode LocalID` прописывается в зависимости от команды `crypto isakmp identity`:

   - Если `crypto isakmp identity address` (вариант по умолчанию), а также в случае `crypto isakmp identity dn` (вариант, неприменимый для

preshared keys), LocalID в конфигурацию не пишется (обозначает – использовать локальный IP-адрес).

- Если crypto isakmp identity hostname, пишется:
  LocalID = IdentityEntry(KeyID *= "<local_id>")
  где <local_id> – представление в виде Hex-string полного DNS-адреса, составленного из локального hostname, заданного с помощью команды hostname <…> и доменного имени, заданного с помощью команды ip domain name <…>. Если доменное имя отсутствует, или в hostname присутствует хотя бы одна точка, <local_id> составляется только из hostname.

RemoteID прописывается по следующим правилам:

- Если ключ привязан к IP-адресу с помощью команды
  crypto isakmp key <…> address <…>,
  то формируется правило с аутентификацией по данному ключу:

```
AuthMethodPreshared <...> (
    [ LocalID  = IdentityEntry( KeyID *= "<local_id>" ) ] –
    если необходимо
    RemoteID = IdentityEntry( IPv4Address *= <peer_ip> )
    SharedIKESecret = <...>
)
```

- Если ключ привязан к hostname с помощью команды
  crypto isakmp key <…> hostname <…>, а hostname привязан к IP-адресу с помощью команды ip host <hostname> <ip-addr>, то формируется правило с аутентификацией по hostname и IP-address:

```
AuthMethodPreshared <...> (
    [ LocalID  = IdentityEntry( KeyID *= "<local_id>" ) ] –
    если необходимо
    RemoteID = IdentityEntry(
    IPv4Address *= <peer_ip> KeyID *= "<peer_id>" )
    SharedIKESecret = <...>
)
```

  где <peer_id> – представление в виде Hex-string hostname-а из команды crypto isakmp key <…> hostname <…>.

- Если ключ привязан к hostname с помощью команды
  crypto isakmp key <…> hostname <…> и используется динамический crypto map, формируется правило с аутентификацией по hostname:

```
AuthMethodPreshared <...> (
    [ LocalID  = IdentityEntry( KeyID *= "<local_id>" ) ] –
    если необходимо
    RemoteID = IdentityEntry( KeyID *= "<peer_id>" )
    SharedIKESecret = <...>
)
```

- Если удается подобрать дополнительные IP-адреса и hostname, для которых задаются те же самые ключи, тогда эти IP-адреса и KeyID (из hostname-ов) добавляются в RemoteID.

7. Если подобрано правило RSA sig – прописывается правило аутентификации в виде:

```
{ AuthMethodRSASign | AuthMethodGostSign } auth_ca(
    LocalID        =  IdentityEntry( ... )
    [ RemoteID       =  IdentityEntry( ... ) ] |
    [ DoNotMapRemoteIDToCert = TRUE ]
    AcceptCredentialFrom      *= CertDescription(
```

```
        X509IssuerDN  *= "..."
        SerialNumber  =  "..."
        X509SubjectDN *= "..."
    )
    SendRequestMode =  { AUTO | NEVER | ALWAYS }
    SendCertMode    =  { AUTO | NEVER | ALWAYS }
)
ModeAuthMethod *= auth_ca
```

- `LocalID для main mode` формируется по следующим правилам:
  - Если `crypto isakmp identity hostname`, пишется:
    `LocalID =  IdentityEntry( FQDN* = USER_SPECIFIC_DATA )`
  - Если `crypto isakmp identity dn:`
    `LocalID =IdentityEntry( DistinguishedName* = USER_SPECIFIC_DATA )`
  - Если `crypto isakmp identity address:`
    `LocalID = IdentityEntry(IPv4Address* = USER_SPECIFIC_DATA)`
- `RemoteID` формируется по следующим правилам:
  - Если к crypto map привязана identity, в которой прописан один или несколько dn, пишется:

```
RemoteID = IdentityEntry (
  DistinguishedName* = CertDescription(Subject* =“<dn1>”),
                       CertDescription( Subject* =“<dn2>”),
...
  FQDN* = “<fqdn1>”, “<fqdn2>” ...
)
```

  - Если к crypto map не привязана `identity`, то пишется
    `DoNotMapRemoteIDToCert = TRUE`

    данная логика предназначена для того, чтобы в качестве remote identity работал IP-адрес и в том случае, когда он отсутствует в сертификате (типичная ситуация). Это бывает полезно при использовании разных типов аутентификации в пределах одной конфигурации - на сертификатах и на preshared keys.

8. В зависимости от наличия команды `crypto isakmp keepalive` и ее параметров в `IKERule` прописываются настройки `DPD`:

   Если команда `crypto isakmp keepalive` не задана (по умолчанию), прописывается `DoNotUseDPD = TRUE`.

   Если команда задана, пишутся параметры:

```
DPDIdleDuration = <secs>
DPDResponseDuration = <retries>
DPDRetries = <dpd_retries>
```

   где `<secs>` – первый аргумент команды;
   `<retries>` – второй аргумент команды;
   `<dpd_retries>` – параметр `dpd_retries` из файла `cs_conv.ini`.
   См. описание настройки `dpd_retries`.

   Внимание!!! Поведение по умолчанию отличается от настроек DPD в версии 2.0. Там всегда (вне зависимости от `crypto isakmp keepalive`) выставлялись настройки DPD по умолчанию, характерные для Native-LSP: `DPDIdleDuration`

= 60; DPDResponseDuration = 5; DPDRetries = 3.
Сейчас по умолчанию DPD выключен.

9. Для сочетания crypto map и интерфейса запоминается сформированное поле Action в структуре FilteringRule. Если данное сочетание снова встречается при обработке другого фильтра – сразу прописывается запомненная строка Action.
10. В конце конвертирования делается попытка загрузить сформированную конфигурацию.

    Если конфигурацию не удалось загрузить, она сохраняется в файле erroneous_lsp.txt (с выдачей сообщения в лог). Файл расположен в:

    Windows – в каталоге агента.

    Solaris и Linux – в каталоге /var/cspvpn.

Внутренние настройки консоли и конвертора

1. Внутренние настройки конвертора хранятся в файле cs_cons_reg.ini, расположенном в каталоге агента.
2. Данный файл используется для хранения внутренних настроек консоли и конвертора. Он автоматически модифицируется при запуске консоли. Редактирование этого файла вручную не рекомендуется.
3. Если файл отсутствует на момент старта консоли, он автоматически создается.
4. Формат файла:
   - Обычный текстовый файл в кодировке ASCII. Используется кодирование окончания строк принятое для операционной системы (Windows/UNIX).
   - Пустые строки и строки, начинающиеся с ! (восклицательный знак) игнорируются.
   - Файл состоит из секций. Каждая секция начинается с названия секции, заключенного в квадратные скобки.
5. В настоящее время присутствует одна секция: описание перекодировки интерфейсов из формата Cisco в Native формат агента:

   Название секции: [interface_list]

   Формат строк: <Native_interface_name> = <Cisco_interface_name>. Например: I0 = 0/0

   Данная секция редактируется автоматически при старте консоли:

   - Если в Cisco-like конфигурации и в INI-файле не описан native interface (например, первый старт консоли или данный native interface был добавлен с помощью if_mgr между двумя стартами консоли), то этому интерфейсу присваивается свободное <Cisco_interface_name>. Этот интерфейс добавляется в Cisco-like конфигурацию и в INI-файл.
   - Если здесь описан native interface, который отсутствует в агенте (например, был удален с помощью if_mgr между двумя запусками консоли), то этот интерфейс удаляется как из INI-файла, так и из Cisco-like конфигурации.
   - Если в текущей Cisco-like конфигурации присутствует интерфейс, не описанный в INI-файле (нештатная ситуация), этот интерфейс удаляется из Cisco-like конфигурации.
   - Если в INI-файле присутствует интерфейс, которого нет в текущей Cisco-like конфигурации (нештатная ситуация), этот интерфейс удаляется из INI-файла.
6. В случае, если произошли какие-либо изменения, описанные в предыдущем пункте, то обновленный файл сохраняется. Если сохранение по тем или иным причинам не удалось, то в лог выдается сообщение об ошибке [3.12].

Управление конвертором с помощью INI-файла

1. Настройки конвертора хранятся в INI-файле `cs_conv.ini`, расположенном в каталоге агента.
2. INI-файл хранит служебную информацию, необходимую для конвертора, включающую в себя все пользовательские настройки.
3. Формат файла:
   - обычный текстовый файл в кодировке ASCII. Используется кодирование окончания строк, принятое для операционной системы (Windows/UNIX)
   - пустые строки и строки, начинающиеся с ! (восклицательный знак) игнорируются
   - файл состоит из нескольких секций. Каждая секция начинается с названия секции, заключенного в квадратные скобки. Например: `[interface_list]`.
4. Описание секций файла:
   - Описание отдельных глобальных настроек:
     - Название секции: `[global_settings]`
     - Формат строк:
       - `product_type = {SERVER | GATE}` – тип агента. Пользователю не следует менять данный параметр.
       - `ike_autopass = { on|off }` – включение/выключение прописывания ike autopass в конфигурации. По умолчанию – on. Пользователю редактировать данный параметр не рекомендуется.
       - `dpd_retries = {1 – 10}` – количество попыток проведения DPD-обмена. По умолчанию – 5. Пользователь может настраивать данный параметр для получения оптимального количества DPD-retries.
       - `tunnel_local_ip = {on | off}` – включение/выключение прописывания локального IP-адреса в TunnelEntry. По умолчанию – `off`. Пользователю редактировать данный параметр не рекомендуется: только при возникновении ситуаций, когда прописывание локального адреса необходимо (пока не выявлено).
       - `policy_sync = = { on | off }` – включение/выключение режима синхронизации политик. (См. п. 4 раздела "Логика запуска конвертора"). По умолчанию – `off`. Пользователь может по своему усмотрению выключить данный параметр, если для него удобнее соответствующее поведение консоли.
   - Описание перекодировки алгоритмов:
     - Название секции: `[algorithm_list]`
     - Формат строк: `<Generic_algorithm_name> = <Native_algorithm_name>`. В качестве `<Generic _algorithm_name>` могут использоваться следующие алгоритмы (регистр важен): `ike-hash-md5, ah-integrity-md5, esp-integrity-md5, esp-cipher-des, ike-cipher-des, ike-hash-sha1, ah-integrity-sha1, esp-integrity-sha1, esp-cipher-3des, esp-cipher-aes, esp-cipher-aes-192, esp-cipher-aes-256, esp-cipher-null, ike-cipher-3des, ike-cipher-aes, ike-cipher-aes-192, ike-cipher-aes-256, ike-group-vko, ike-group-1, ike-group-2, ike-group-5, pfs-group-vko, pfs-group-group1, pfs-group-group2, pfs-group-group5. Например: ike-cipher-3des=DES3-K168-CBC`

- При необходимости использовать алгоритм VKO совместно со средствами управления Cisco (CSM, CiscoWorks) надо скорректировать файл cs_conv.ini:
  - Выбрать неиспользуемую в IKE Diffie-Helman группу.
    Рекомендуется выбирать минимальную неиспользуемую группу:
    Для полностью новой инфраструктуры – группу 1. Если в существующей инфраструктуре группа 1 уже используется, то группа 2 и т.п.
  - Для выбранной группы в файле cs_conv.ini прописать конвертирование в алгоритм VKO_1B.
  - Аналогично выбрать неиспользуемую в PFS Diffie-Helman группу.
    Для простоты рекомендуется выбрать ту же самую группу, что и для IKE.
  - Для выбранной группы в файле cs_conv.ini прописать конвертирование в алгоритм VKO_1B.

    Пример строк, исправленных для использования алгоритма VKO совместно со средствами управления Cisco:

```
ike-group-vko = VKO_1B
ike-group-1 = VKO_1B
ike-group-2 = MODP_1024
ike-group-5 = MODP_1536
pfs-group-vko = VKO_1B
pfs-group-group1 = VKO_1B
pfs-group-group2 = MODP_1024
pfs-group-group5 = MODP_1536
```

  - Редактирование пользователем остальных параметров данной секции, как правило, не требуется, но возможно при необходимости перенести перекодировку алгоритмов ГОСТ на другие алгоритмы, или для полного отказа от перекодировки ГОСТ на агенте.
- Описание логики работы с `Cert request` и посылки сертификатов в процессе IKE:
  - Название секции: `[auth_cert]`
  - Формат строк: `<param_name>=<param_val>`. Список названий `<param_name>`:

    `send_request`. По умолчанию – ALWAYS.
    `send_cert`. По умолчанию – ALWAYS.
  - Редактирование пользователем данной секции, как правило, не требуется, но возможно при необходимости изменить логику работы с Cert request и посылки сертификатов в процессе IKE.
- Описание переменных окружения, выставляемых для процесса cs_console:
  - Название секции: `[env]`
  - Формат строк: `<env_var_name>=<env_var_val>`

    Можно задавать любые перемнные окружения. На практике, как правило, нужно для выставления переменной PATH.
  - Редактирование пользователем данной секции, как правило, не требуется, но возможно при необходимости изменить переменную PATH или добавить какие-то свои переменные окружения (м.б. полезно для команды run).

5. По умолчанию в `cs_conv.ini` используется следующая подстановка: `MD5 и DES` заменяются на алгоритмы КриптоПРО (как в версии для КриптоПРО, так и в версии для СигналКОМ).
   - Только в версии для СигналКОМ: Рядом с основным файлом `cs_conv.ini` присутствуют еще два файла:

- `cs_conv.ini.legacy` – настройки, при которых можно обеспечить соединение на ГОСТ-алгоритмах, используемых в agent 2.0 sc (для совместимости с этой версией агента). При этом через `cs_console` нельзя использовать КриптоПРО алгоритмы. Заменяются `MD5 и DES`.
- `cs_conv.ini.mixed` – настройки, которые позволяют обеспечивать соединения по ГОСТ-алгоритмам как используемым в agent 2.0 sc, так и по КриптоПРО алгоритмам. MD5 и DES заменяются на КриптоПРО алгоритмы, а SHA1 и 3DES – на алгоритмы, используемые в agent 2.0 sc. Следует отметить, что при этом вообще невозможно установить соединения не по ГОСТ-овым алгоритмам (в частности, невозможно соединение с устройствами Cisco).

6. Варианты файлов, поставляемых в составе продукта:

   Пример INI-файла **cs_conv.ini** для Gate (СигналКОМ и КриптоПРО, Solaris):

```
[global_settings]
ike_autopass    = on
dpd_retries     = 5
product_type    = GATE
tunnel_local_ip = off
policy_sync     = on

 [algorithm_list]

! Original algorithms:

! ike-hash-md5        = MD5
! ah-integrity-md5    = MD5-H96-HMAC
! esp-integrity-md5   = MD5-H96-HMAC
! esp-cipher-des      = DES-CBC
! ike-cipher-des      = DES-CBC

! Replaced algorithms:

ike-hash-md5        = GR341194CPRO1-65534
ah-integrity-md5    = GR341194CPRO1-H96-HMAC-254
esp-integrity-md5   = GR341194CPRO1-H96-HMAC-65534
esp-cipher-des      = G2814789CPRO1-K256-CBC-254
ike-cipher-des      = G2814789CPRO1-K256-CBC-65534

ike-hash-sha1       = SHA1
ah-integrity-sha1   = SHA1-H96-HMAC
esp-integrity-sha1  = SHA1-H96-HMAC
esp-cipher-3des     = DES3-K168-CBC
ike-cipher-3des     = DES3-K168-CBC
esp-cipher-aes      = AES-K128-CBC-12
esp-cipher-aes-192  = AES-K192-CBC-12
```



```
esp-cipher-aes-256  = AES-K256-CBC-12
esp-cipher-null     = NULL
ike-cipher-aes      = AES-K128-CBC-7
ike-cipher-aes-192  = AES-K192-CBC-7
ike-cipher-aes-256  = AES-K256-CBC-7
ike-group-vko       = VKO_1B
ike-group-1         = MODP_768
ike-group-2         = MODP_1024
ike-group-5         = MODP_1536
pfs-group-vko       = VKO_1B
pfs-group-group1    = MODP_768
pfs-group-group2    = MODP_1024
pfs-group-group5    = MODP_1536

[auth_cert]
send_request            = ALWAYS
send_cert               = ALWAYS

[env]
PATH=/usr/sbin:/usr/bin
```

Пример INI-файла **cs_conv.ini** для Gate (СигналКОМ и КриптоПРО, Linux):

```
[global_settings]
ike_autopass    = on
dpd_retries     = 5
product_type    = GATE
tunnel_local_ip = off
policy_sync     = on

 [algorithm_list]

! Original algorithms:

! ike-hash-md5        = MD5
! ah-integrity-md5    = MD5-H96-HMAC
! esp-integrity-md5   = MD5-H96-HMAC
! esp-cipher-des      = DES-CBC
! ike-cipher-des      = DES-CBC

! Replaced algorithms:

ike-hash-md5        = GR341194CPRO1-65534
ah-integrity-md5    = GR341194CPRO1-H96-HMAC-254
```

```
esp-integrity-md5   = GR341194CPRO1-H96-HMAC-65534
esp-cipher-des      = G2814789CPRO1-K256-CBC-254
ike-cipher-des      = G2814789CPRO1-K256-CBC-65534

ike-hash-sha1       = SHA1
ah-integrity-sha1   = SHA1-H96-HMAC
esp-integrity-sha1  = SHA1-H96-HMAC
esp-cipher-3des     = DES3-K168-CBC
ike-cipher-3des     = DES3-K168-CBC
esp-cipher-aes      = AES-K128-CBC-12
esp-cipher-aes-192  = AES-K192-CBC-12
esp-cipher-aes-256  = AES-K256-CBC-12
esp-cipher-null     = NULL
ike-cipher-aes      = AES-K128-CBC-7
ike-cipher-aes-192  = AES-K192-CBC-7
ike-cipher-aes-256  = AES-K256-CBC-7
ike-group-vko       = VKO_1B
ike-group-1         = MODP_768
ike-group-2         = MODP_1024
ike-group-5         = MODP_1536
pfs-group-vko       = VKO_1B
pfs-group-group1    = MODP_768
pfs-group-group2    = MODP_1024
pfs-group-group5    = MODP_1536

[auth_cert]
send_request            = ALWAYS
send_cert               = ALWAYS

[env]
PATH=/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin
```

Пример **cs_conv.ini.legacy** для Gate (только СигналКОМ):

```
! Agent 2.0 sc compatible algorithms support.
[global_settings]
ike_autopass    = on
dpd_retries     = 5
product_type    = GATE
tunnel_local_ip = off
policy_sync     = on

[algorithm_list]

! Original algorithms:
```

```
! ike-hash-md5        = MD5
! ah-integrity-md5    = MD5-H96-HMAC
! esp-integrity-md5   = MD5-H96-HMAC
! esp-cipher-des      = DES-CBC
! ike-cipher-des      = DES-CBC

! Replaced algorithms:

ike-hash-md5        = SCR341194-65533
ah-integrity-md5    = SCR341194-H96-HMAC-253
esp-integrity-md5   = SCR341194-H96-HMAC-65533
esp-cipher-des      = SCG2814789-K256-CBC-253
ike-cipher-des      = SCG2814789-K256-CBC-65533

ike-hash-sha1       = SHA1
ah-integrity-sha1   = SHA1-H96-HMAC
esp-integrity-sha1  = SHA1-H96-HMAC
esp-cipher-3des     = DES3-K168-CBC
ike-cipher-3des     = DES3-K168-CBC
esp-cipher-aes      = AES-K128-CBC-12
esp-cipher-aes-192  = AES-K192-CBC-12
esp-cipher-aes-256  = AES-K256-CBC-12
esp-cipher-null     = NULL
ike-cipher-aes      = AES-K128-CBC-7
ike-cipher-aes-192  = AES-K192-CBC-7
ike-cipher-aes-256  = AES-K256-CBC-7
ike-group-vko       = VKO_1B
ike-group-1         = MODP_768
ike-group-2         = MODP_1024
ike-group-5         = MODP_1536
pfs-group-vko       = VKO_1B
pfs-group-group1    = MODP_768
pfs-group-group2    = MODP_1024
pfs-group-group5    = MODP_1536

[auth_cert]
send_request            = ALWAYS
send_cert               = ALWAYS

[env]
PATH=<Зависит от ОС. См. выше>
```


Пример **cs_conv.ini.mixed** для Gate (только СигналКОМ):

```
! Agent 2.0 sc compatible and current algorithms support.
[global_settings]
ike_autopass    = on
dpd_retries     = 5
product_type    = GATE
tunnel_local_ip = off
policy_sync     = on

[algorithm_list]

! Original algorithms:

! ike-hash-md5        = MD5
! ah-integrity-md5    = MD5-H96-HMAC
! esp-integrity-md5   = MD5-H96-HMAC
! esp-cipher-des      = DES-CBC
! ike-cipher-des      = DES-CBC
! ike-hash-sha1       = SHA1
! ah-integrity-sha1   = SHA1-H96-HMAC
! esp-integrity-sha1  = SHA1-H96-HMAC
! esp-cipher-3des     = DES3-K168-CBC
! ike-cipher-3des     = DES3-K168-CBC

! Replaced algorithms:

ike-hash-md5        = GR341194CPRO1-65534
ah-integrity-md5    = GR341194CPRO1-H96-HMAC-254
esp-integrity-md5   = GR341194CPRO1-H96-HMAC-65534
esp-cipher-des      = G2814789CPRO1-K256-CBC-254
ike-cipher-des      = G2814789CPRO1-K256-CBC-65534
ike-hash-sha1       = SCR341194-65533
ah-integrity-sha1   = SCR341194-H96-HMAC-253
esp-integrity-sha1  = SCR341194-H96-HMAC-65533
esp-cipher-3des     = SCG2814789-K256-CBC-253
ike-cipher-3des     = SCG2814789-K256-CBC-65533

esp-cipher-aes      = AES-K128-CBC-12
esp-cipher-aes-192  = AES-K192-CBC-12
esp-cipher-aes-256  = AES-K256-CBC-12
esp-cipher-null     = NULL
ike-cipher-aes      = AES-K128-CBC-7
ike-cipher-aes-192  = AES-K192-CBC-7
ike-cipher-aes-256  = AES-K256-CBC-7
```



```
ike-group-vko       = VKO_1B
ike-group-1         = MODP_768
ike-group-2         = MODP_1024
ike-group-5         = MODP_1536
pfs-group-vko       = VKO_1B
pfs-group-group1    = MODP_768
pfs-group-group2    = MODP_1024
pfs-group-group5    = MODP_1536

[auth_cert]
send_request            = ALWAYS
send_cert               = ALWAYS

[env]
PATH=<Зависит от ОС. См. выше>
```

Сообщения в логе при конвертировании

При работе конвертора могут посылаться сообщения в Syslog.

Формат строк сообщений:

`<Date_Time> <Level:> <Message>, где Level – INFO, Warning или ERROR.`

Пример сообщения:

`Wed Oct 29 18:19:50 2003 INFO: LSP conversion complete. Warnings: 2`

Список сообщений, предупреждений и ошибок, выдаваемых в логе, представлен в таблице.

1. **Информационные сообщения**

	Сообщение	Комментарий
1.1	LSP conversion started	Начат процесс конвертирования
1.2	LSP conversion complete	Процесс конвертирования завершен успешно. Предупреждения не выдавались.
1.3	LSP conversion complete. Warnings: {1}	Процесс конвертирования завершен успешно. Выдано {1} предупреждений.
1.4	Host mode is enabled.	Включен Host-режим
1.5	Previous user-defined LSP saved in file "{1}"	Предыдущая пользовательская LSP сохранена в файле “{1}”
1.6	Non-synchronized policy detected. Policy type: <type> где <type> один из: DDP	Обнаружена несинхронизированная политика. Тип политики: <type>

	Drop All User-defined (Source: <source>), где <source> – Agent или Command-line utility.	
1.7	Incremental policy loading disabled by policy_sync setting (file cs_conv.ini)	Инкрементальная политика отключена из-за настройки policy_sync (файл cs_conv.ini)
1.8	Incremental policy loading disabled due to policy synchronization fail	Инкрементальная политика отключена из-за того, что не удалось провести синхронизацию политик

2. **Предупреждения**

	Сообщение	Комментарий
2.1	LDAP url "{1}" ignored. IP address and port allowed only.	Введенный LDAP url {1} проигнорирован, поскольку допускаются только IP-адрес и порт.
2.2	OUT access group in the interface "{1}" ignored. Only IN access group is used.	Проигнорирован access-group out в интерфейсе {1}, поскольку допускается только access-group in.
2.3	Only one interface is used while host mode is on. Other interfaces ignored.	При включенном Host-режиме допускается только один интерфейс. Остальные интерфейсы игнорируются.
2.4	Only one CA certificate imported. Other certs ignored.	Импортирован только первый по списку CA-сертификат. End-User сертификаты и оставшиеся CA-сертификаты проигнорированы.
2.5	Crypto map "{1}" contains several peers. Peer(s) "{2}" ignored due to authentication information mismatch.	В crypto map {1} прописаны несколько peer-ов. Peer(s) {2} проигнорированы из-за того, что для них не совпадает аутентификационная информация.
2.6	Crypto map(s) "{1}" contain transform sets with different encapsulation modes. Tunnel mode is used.	Crypto maps {1} содержат transform sets, в которых заданы разные encapsulation режимы. Используется туннельный режим.
2.7	Crypto map(s) "{1}" contain transform sets with different encapsulation modes. Transport mode is used.	Crypto maps {1} содержат transform sets, в которых заданы разные encapsulation режимы. Используется транспортный режим.
2.8	Incorrect config detected. Policy conversion ignored	Обнаружена некорректная политика. Конвертирование политики не делается.
2.9	Crypto map set(s) "{1}" contain static crypto map(s) with priorities lower than dynamic.	Crypto map set(s) {1} содержат статические crypto map(s) с приоритетом ниже, чем у динамических
2.10	Crypto map "{1}" contains several peers with different preshared keys. This is not recommended.	Crypto map {1} содержит несколько peers с разными preshared keys. Это не рекомендуемая ситуация. Подробнее см. п.8 для несовпадающих Preshared keys.

3. **Ошибки**

	Сообщение	Комментарий
3.1	Cannot read settings form INI file. Conversion failed.	Невозможно прочитать настройки из INI-файла.
3.2	No interfaces were found in the INI file. Configure interfaces or set host mode to proceed.	Не заданы интерфейсы в INI-файле при выключенном Host-режиме. Необходимо настроить интерфейсы или включить Host-режим.
3.3	No interfaces were found in the configuration.	В импортируемой конфигурации не заданы интерфейсы. Конвертирование не имеет смысла.
3.4	Interface "{1}" not found in the INI file. Conversion aborted.	Интерфейс {1} не задан в INI-файле. Конвертирование остановлено.
3.5	Certificate parse failed	Не удалось разобрать введенный сертификат.
3.6	Could not convert crypto map “{1}”. Reason: <Reason> где <Reason>: There is no isakmp policy. There is no CA or appropriate preshared key. Also isakmp policy can have wrong type (rsa-sig or pre-share). There is no peer. There are no transform sets. Crypto map is incomplete. Unknown.	Невозможно сконвертировать crypto map “{1}”. Причина: <Причина> где <Причина> одна из: Отсутствует isakmp policy. Отсутствует CA или подходящий Preshared Key, либо isakmp policy неправильного типа (rsa-sig или pre-share). Отсутствует peer. Отсутствуют transform sets Crypto map неполная (не хватает crypto ACL, transform set или peer). Неизвестная причина.
3.7	LSP load failed	Не удалось загрузить сформированную LSP
3.8	Unsupported network wildcard "{1}"	Не поддерживается данный формат маски подсети
3.9	LSP conversion failed	Произошла некоторая невыясненная ошибка
3.101	Could not save previous user-defined LSP in file "{1}"	Не удалось сохранить предыдущую пользовательскую LSP в файл {1}
3.11	Could not save internal settings in file "{1}"	Не удалось сохранить внутренние настройки в файл {1}
3.12	Address pool "{1}" not found. Conversion aborted.	Не найден пул адресов “{1}”. Конвертирование прервано.

Описание обработки интерфейсов

1. Из интерфейса читается `access list`, прописанный в команде `ip access-group <access_list> in` (режим настройки интерфейса).
   - Далее для простоты такой `access list` будет указываться как `acl-in`.
   - Если такая команда не вводилась, то считается, что прописан `access list` с неявным правилом `Pass All` (на интерфейсе).
   - Поскольку в данном `access list` прописывается условно входящий трафик, поле `Source` транслируется в поле `PeerIPFilter`, а поле `Destination – LocalIPFilter`.
   - В поле `NetworkInterfaces` структуры `FilteringRule` прописывается внутреннее ("агентское") имя текущего интерфейса (см. Описание перекодировки алгоритмов).
   - Если в данной команде стоит ссылка на `access list`, то в конце данного `access list` предполагается неявное правило `Drop All`.

   Пример

   ```
   Interface FastEthernet0/0
   ip address 1.1.1.1 255.255.0.0
   ip access-group acl-in in
   ...
   Exit
   ```

2. Из интерфейса последовательно читаются `crypto maps` из `crypto map set`, прописанного в команде `crypto map <crypto_map>` (конфигурационный режим интерфейса).
   - Из описания `crypto map` читается `access list`, прописанный в команде `match address <access_list>` (конфигурационный режим `crypto map`).
   - Далее для простоты такой `access list` будет указываться как `crypto-map-acl`.
   - Если такой `access-list` не прописан, то считается, что прописан `access list` с неявным правилом `Pass All`.
   - Поскольку в `crypto-map-acl` прописывается условно выходящий трафик, трансляция адресов производится зеркально по отношению к `acl-in`.

   Пример (для интерфейса и crypto map прописываются фактически одинаковые access lists):

   ```
   ip access-list ex acl-in
   permit udp 1.1.1.1 0.0.0.0 2.2.2.2 0.0.0.0
   exit

   ip access-list ex crypto-map-acl
   permit udp 2.2.2.2 0.0.0.0 1.1.1.1 0.0.0.0
   exit

   crypto map cr-map 1 ipsec-isakmp
   ```

```
...
match address crypto-map-acl
exit

Interface FastEthernet0/0
...
ip access-group acl-in in
crypto map cr-map
...
exit
```

3. Если никакой `crypto map` не привязан к интерфейсу:

   - Происходит однозначное перекодирование из `acl-in` в Native фильтры: `deny -> DROP, permit -> PASS.`

   Пример

   # Cisco-like конфигурация:

```
!...
ip access-list ex acl-1
deny udp 1.1.1.1 0.0.0.0 1.1.1.2 0.0.0.0
permit 1 1.1.1.1 0.0.0.0 1.1.1.2 0.0.0.0
exit
!
Interface FastEthernet0/0
ip address 1.1.1.2 255.255.0.0
ip access-group acl-1 in
exit
!...
```

   # Native-LSP конфигурация:

```
...

FilteringRule acl_1
(
    LocalIPFilter *= FilterEntry( IPAddress *= 1.1.1.2 ProtocolID *= 17 )
    PeerIPFilter  *= FilterEntry( IPAddress *= 1.1.1.1 ProtocolID *= 17 )
    NetworkInterfaces *= "if0"
    Action *= ( DROP )
)

FilteringRule acl_1_1
(
    LocalIPFilter *= FilterEntry( IPAddress *= 1.1.1.2 ProtocolID *= 1 )
    PeerIPFilter  *= FilterEntry( IPAddress *= 1.1.1.1 ProtocolID *= 1 )
```

```
    NetworkInterfaces *= "if0"
    Action *= ( PASS )
)

FilteringRule acl_1_2
(
    LocalIPFilter *= FilterEntry( IPAddress *= 0.0.0.0..255.255.255.255 )
    PeerIPFilter  *= FilterEntry( IPAddress *= 0.0.0.0..255.255.255.255 )
    Action *= ( DROP )
    NetworkInterfaces *= "if0"
)
# ...
```

4. Если `crypto map` присутствует, то происходит чтение правил в `acl-in`:
   - Правило `deny` напрямую перекодируется в `DROP`.
   - В случае правила `permit` делается проход по `crypto-map-acl`:
     - Берется правило из `crypto-map-acl`. Сравнивается адресная информация из правила `acl-in` с адресной информацией в правиле `crypto-map-acl` с учетом смены `source` и `destination` (например: `1.1.1.0 0.0.0.255 range 10 20 2.2.2.0 0.0.0.255 -> 2.2.2.0 0.0.0.255 1.1.1.0 0.0.0.255 range 10 20`). Делается попытка определить пересечение этих подмножеств адресов (частными случаями пересечения являются также полное совпадение и включение одного из подмножеств в другое). В случае удачного сравнения формируется адрес `work_address`, содержащий в себе пересечение подмножеств адресов. Далее, для определенности, предполагается, что в `work_address` используется порядок `source/destination`, как в `crypto-map-acl`.

Примеры

acl-in address	crypto-map-acl address	work_address
tcp host 1.1.1.1 any	ip any any	tcp any host 1.1.1.1
udp 1.1.1.0 0.0.0.255 eq 10 2.2.2.0 0.0.0.255 range 10 50	udp host 1.1.1.1 eq 10 host 2.2.2.2 range 20 30	udp host 1.1.1.1 eq 10 host 2.2.2.2 range 20 30
udp host 1.1.1.1 host 2.2.2.2	udp host 1.1.1.1 host 2.2.2.2	udp host 1.1.1.1 host 2.2.2.2

     - Если не удалось определить пересечение адресов (иначе говоря, нулевое пересечение); тогда правило из `crypto-map-acl` игнорируется.
     - Если удалось определить пересечение адресов и сформировать `work_address`, прописывается правило с адресной информацией из `work_address`:
       - Если в правиле `crypto-map-acl` прописан `deny -> правило PASS`.
       - Если в правиле `crypto-map-acl` прописан `permit -> правило APPLY (IPsec)`. При этом пишутся параметры из данного `crypto map`.

- В конце прохода по `crypto map` прописывается правило `PASS` с адресной информацией из правила из `acl-in`.

5. В случае, если в `crypto map set` присутствует ссылка на `dynamic template set` (задается командами `crypto dynamic map`), в котором есть несколько `dynamic crypto maps`, в `crypto-map-acls` которых существуют пересечения по адресам, в `FilteringRule` происходит объединение правил.

Пример

#Фрагмент Cisco-like конфигурации:

```
ip access-list extended a1
 permit  icmp  192.168.101.0 0.0.0.255 10.0.12.240 0.0.0.15
!
crypto dynamic-map dmap 1
 match address a1

...

crypto dynamic-map dmap 2
 match address a1

...
crypto map cmap 1 ipsec-isakmp dynamic dmap

interface FastEthernet0/0
 crypto map cmap
```

#Фрагмент Native-LSP:

```
FilteringRule Filter_nil_acl_dmap_1
(
    LocalIPFilter *= FilterEntry(
        IPAddress *= 192.168.101.0/24 ProtocolID *= 1 )
    PeerIPFilter  *= FilterEntry(
        IPAddress *= 10.0.12.240/28 ProtocolID *= 1 )
    NetworkInterfaces *= "I0"
    Action *= ( dmap_1 ), ( dmap_2 )
)
```

- Объединение правил для статических `crypto maps` не производится (ни между разными статическими `crypto maps`, ни между статическими и динамическими `crypto maps`).
- В случае, если статическая `crypto map` имеет приоритет ниже, чем динамическая, то могут возникать логические неувязки. Настоятельно рекомендуется давать статическим `crypto maps` приоритет выше, чем динамическим. Следует отметить, что в документации Cisco также присутствует эта рекомендация.

- Если данная рекомендация не выполнена – выдается предупреждение [2.9].

- Не производится объединение правил для динамических `crypto maps`, которые входят в разные `dynamic template sets`, которые в свою очередь входят в один `crypto map set`.

Пример

#Фрагмент Cisco-like конфигурации:

```
ip access-list extended a1
 permit  icmp  192.168.101.0 0.0.0.255 10.0.12.240 0.0.0.15
!
crypto dynamic-map dmap1 1
match address a1
...
crypto dynamic-map dmap1 2
match address a1
...
crypto dynamic-map dmap2 1
match address a1
...
crypto map cmap 1 ipsec-isakmp dynamic dmap1

crypto map cmap 2 ipsec-isakmp dynamic dmap2

interface FastEthernet0/0
 crypto map cmap
```

#Фрагмент Native-LSP

(dmap2 не попала в конфигурацию, поскольку объединение правил для нее не выполнялось, а сформированный фильтр не был прописан, поскольку полностью совпал с фильтром, который был прописан ранее; подробнее см. ниже):

```
FilteringRule Filter_nil_acl_dmap1_1
(
    LocalIPFilter *= FilterEntry(
        IPAddress *= 192.168.101.0/24 ProtocolID *= 1 )
    PeerIPFilter  *= FilterEntry(
        IPAddress *= 10.0.12.240/28 ProtocolID *= 1 )
    NetworkInterfaces *= "I0"
    Action *= ( dmap1_1 ), ( dmap1_2 )
)
```

В случае, если в `dynamic template set` существует пересечение по адресам правил, в которых для одних `dynamic templates` прописаны правила `permit`, а для других – `deny`, в `FilteringRule` прописывается правило вида `(PASS), (Action1), …, (ActionN).`

Пример

#Фрагмент Cisco-like конфигурации:

```
ip access-list extended a1
 permit  icmp  192.168.101.0 0.0.0.255 10.0.12.240 0.0.0.15
!
ip access-list extended a2
 deny  icmp  192.168.101.0 0.0.0.255 10.0.12.240 0.0.0.15
!
crypto dynamic-map dmap 1
 match address a1
...
crypto dynamic-map dmap 2
 match address a2
...
crypto dynamic-map dmap 3
 match address a1
...
crypto map cmap 1 ipsec-isakmp dynamic dmap
...
interface FastEthernet0/0
 crypto map cmap
```

#Фрагмент Native-LSP:

```
FilteringRule Filter_nil_acl_dmap_1
(
    LocalIPFilter *= FilterEntry(
       IPAddress *= 192.168.101.0/24 ProtocolID *= 1 )
    PeerIPFilter  *= FilterEntry(
       IPAddress *= 10.0.12.240/28 ProtocolID *= 1 )
    NetworkInterfaces *= "I0"
    Action *= ( PASS ), ( dmap_1 ), ( dmap_3 )
)
```

- Логика формирования данных фильтров может существенно отличаться от логики Cisco.
- В данном примере продемонстрирован особый прием: специально для прописывания `PASS`-правила сделан `crypto dynamic-map dmap 2` (на самом деле приоритет этого `dynamic map` в данном конкретном случае не важен), в котором нет ничего, кроме связи с `ACL`, состоящим из `deny`-правила (правил): отсутствуют `transform sets` и т.п. Следует отметить, что данный способ может использоваться только с агентом, и неприменим на реальных устройствах Cisco.
- Данная логика действует только на явно прописанные `deny`-правила. Для неявных правил `deny ip any any`, которые предполагаются в конце каждого `access list`, никаких объединений правил не делается.

Например, если из предыдущего примера убрать `dmap 2`:

```
ip access-list extended a1
 permit  icmp  192.168.101.0 0.0.0.255 10.0.12.240 0.0.0.15
!
crypto dynamic-map dmap 1
 match address a1
...
crypto dynamic-map dmap 3
 match address a1
...
crypto map cmap 1 ipsec-isakmp dynamic dmap
...
interface FastEthernet0/0
 crypto map cmap
```

#Фрагмент Native-LSP будет уже без PASS-правила:

```
FilteringRule Filter_nil_acl_dmap_1
(
    LocalIPFilter *= FilterEntry(
         IPAddress *= 192.168.101.0/24 ProtocolID *= 1 )
    PeerIPFilter  *= FilterEntry(
         IPAddress *= 10.0.12.240/28 ProtocolID *= 1 )
    NetworkInterfaces *= "I0"
    Action *= ( dmap_1 ), ( dmap_3 )
)
```

6. При формировании `FilteringRule` дополнительно соблюдается следующее правило: для сервера в фильтрах, в которых прописан локальный адрес `ANY`, в Native-LSP прописывается адрес `LOCAL_IP_ADDRESSES`. Для CSP VPN Gate – пишется диапазон `0.0.0.0..255.255.255.255`.
7. Происходит проверка: нужно ли прописывать данный фильтр. Если этот фильтр совпадает или полностью включается в один из предыдущих фильтров, прописанных для данного интерфейса, тогда этот фильтр не прописывается в LSP.

   Пример

   # Cisco-like конфигурация:

```
!...
ip access-list ex crypto-acl-1
deny udp 1.1.1.1 0.0.0.0 eq 500 2.2.2.2 0.0.0.0 eq 500
permit 1 1.1.1.1 0.0.0.0 2.2.2.2 0.0.0.0
exit
!
crypto map crypto-map1 1 ipsec-isakmp
set peer 2.2.2.2
set transform-set transform-1
```

```
match address crypto-acl-1
exit
!
Interface FastEthernet0/0
ip address 1.1.1.1 255.255.0.0
crypto map crypto-map1
exit
!...
```

Native LSP

```
 ...
FilteringRule Filter_nil_acl_crypto_map1_1
(
    LocalIPFilter *= FilterEntry( IPAddress *= 1.1.1.1 ProtocolID *= 17
Port *= 500 )
    PeerIPFilter  *= FilterEntry( IPAddress *= 2.2.2.2 ProtocolID *= 17
Port *= 500 )
    NetworkInterfaces *= "if0"
    Action *= ( PASS )
)

# .IKE & IPsec parameters

IPsecAction crypto_map1_1
(
    TunnelingParameters *= TunnelEntry(
        LocalIPAddress = 1.1.1.1
        PeerIPAddress = 2.2.2.2
        DFHandling=...
    )
    ContainedProposals *= ( ... )
    IKERule = ...
)

FilteringRule Filter_nil_acl_crypto_map1_1_1
(
    LocalIPFilter *= FilterEntry( IPAddress *= 1.1.1.1 ProtocolID *= 1 )
    PeerIPFilter  *= FilterEntry( IPAddress *= 2.2.2.2 ProtocolID *= 1 )
    NetworkInterfaces *= "if0"
    Action *= ( crypto_map1_1 )
)

FilteringRule Filter_nil_acl
(
    LocalIPFilter *= FilterEntry( IPAddress *= 0.0.0.0..255.255.255.255 )
```

```
    PeerIPFilter  *= FilterEntry( IPAddress *= 0.0.0.0..255.255.255.255 )
    NetworkInterfaces *= "if0"
    Action *= ( PASS )
)
# ...
```

Формирование имен структур LSP при конвертировании

1. При конвертировании Cisco-like конфигурации в LSP конфигурацию имена структур LSP формируются из имен и индексов объектов Cisco-like конфигурации. При этом следует учитывать ряд ограничений:
    - В объектах Cisco-like конфигурации разных типов могут использоваться одинаковые имена. В LSP имя объекта должно быть уникальным.
    - Могут использоваться цифровые индексы. В LSP требуется задавать идентификаторы, начинающиеся с буквы.
    - Как правило, синтаксис Cisco-like имен более свободный (например допускаются символы, которые нельзя использовать в идентификаторах LSP).
    - В некоторых случаях требуется формировать имя структуры LSP из группы объектов Cisco-like конфигурации.
    - Один объект Cisco-like конфигурации (или группа объектов) может порождать несколько LSP объектов (каждый из которых должен обладать уникальным именем).
2. Общие сведения по формированию имен:
    - Сначала готовится прототип имени объекта. Для этого прототипа нет каких-то специальных требований: например это может быть имя объекта Cisco-like конфигурации, константная строка, сочетание префикса и имен нескольких объектов и т.п.
    - Далее производится нормализация имени:
        - Все символы, кроме букв латинского алфавита и цифр преобразуются к символу подчеркивания.
        - Если имя начинается с цифры, перед ним ставится буква n.
    - Далее производится поиск полученного имени среди уже сформированных (для обеспечения уникальности):
        - Если имя не найдено, считаем его окончательно сформированным.
        - Если имя найдено, добавляем к нему последовательно суффиксы _1, _2 и т.д. до тех пор, пока не будет найдено имя, которое еще не использовалось.
    - Полученное имя записывается в конфигурацию и запоминается для того, чтобы оно не было использовано для другого объекта.
3. Далее описываются конкретные правила формирования прототипов имен объектов:

Имя структуры	Вариант использования	Правило формирования	Примеры
FilteringRule	Фильтр (без IPsec)	Если на интерфейсе отсутствует фильтрующий ACL, то используется слово "Filter_nil_acl".	Filter_nil_acl
		Если на интерфейсе присутствует фильтрующий ACL, заданный командой access-list, то производится конкатенация (соединение) префикса "Filter" и числового access-list-number из команды access-list.	Filter_101
		Если на интерфейсе присутствует фильтрующий ACL, заданный командой ip access-list – имя ACL, заданное в команде ip access-list.	Filter_acl5
	IPsec, заданный с помощью статической crypto map	Формируется конкатенацией имени FilteringRule без IPsec (см. предыдущий пункт), знака подчеркивания, имени crypto map, знака подчеркивания, индекса crypto map. Примечание: в случае, если в правиле задано несколько правил, имя FilteringRule формируется по первому правилу.	Filter_acl10_cmap_1 Filter_nil_acl_cmap_12
	IPsec, заданный с помощью динамической crypto map	Формируется аналогично, как для статической crypto map. Отличие в том, что вместо имени crypto map используется имя dynamic crypto map template (задается в команде crypto dynamic-map).	Filter_acl5_dmap_2
IKETransform		Конкатенация префикса "IKETransform_" и индекса ISAKMP policy (команда crypto isakmp policy).	IKETransform_10
AHProposal		Конкатенация "AH_" и имени transform-set (команда crypto ipsec transform-set)	AH_trset1
ESPProposal		Конкатенация "ESP_" и имени transform-set (команда crypto ipsec transform-set)	ESP_trset1

Имя структуры	Вариант использования	Правило формирования	Примеры
IKERule	Статическая crypto map	Конкатенация "IKE_", имени crypto map, знака подчеркивания, индекса crypto map.	IKE_cmap_1
	Динамическая crypto map	Формируется аналогично, как для статической crypto map. Отличие в том, что вместо имени crypto map используется имя dynamic crypto map template (задается в команде crypto dynamic-map).	IKE_dmap_2
AuthMethodRSASign AuthMethodDSSSign AuthMethodGOSTSign		auth_ca	auth_ca
AuthMethodPreshared		Конкатенация "IKE_auth_" и имени ключа (как он кладется в базу). Имя ключа формируется как cs_key_<ip_addr> или cs_key_<hostname> (точки заменяются на знак подчеркивания).	IKE_auth_cs_key_192_168_1_2 IKE_auth_cs_key_host1_company_com
CertDescription		ca	ca
IPsecAction	Статическая crypto map	Конкатенация имени crypto map, знака подчеркивания, индекса crypto map.	cmap_1
	Динамическая crypto map	Формируется аналогично, как для статической crypto map. Отличие в том, что вместо имени crypto map используется имя dynamic crypto map template (задается в команде crypto dynamic-map).	dmap_1
AddressPool		Имя pool (команда ip local pool)	pool1

Работа с сертификатами

Регистрация CA сертификата

Зарегистрировать CA сертификат в базе Продукта можно двумя способами:

- с помощью утилиты командной строки ***cert_mgr import***
- через cs_console командами ***crypto pki trustpoint*** и ***crypto pki certificate chain***.

При регистрации сертификата первым способом при первом старте консоли после добавления сертификатов, добавленные сертификаты будут доступны для использования в cisco-like конфигурации. Для них будет создан *trustpoint* с именем *s-terra technological trustpoint*.

Для регистрации CA сертификата через cs_console используются команды:

- ***crypto pki trustpoint*** *name* – для объявления имени CA и входа в режим *ca trustpoint configuration*:

  можно задать несколько таких команд для объявления разных trustpoint

  в режиме этой команды можно указать адрес LDAP-сервера и режимы использования CRL при проверке сертификатов:

  - ***crl query*** *ldap://IP-адрес(:порт)* – задает адрес LDAP-сервера. При обращении к LDAP-серверу шлюз безопасности сначала смотрит поле CDP сертификата, если в этом поле прописанный путь к LDAP-серверу является неполным, то добавляются данные (IP-адрес и порт) из команды *crl query*. Если CDP содержит полный путь, *crl query* не используется. Если в сертификате нет поля CDP, то используется эта команда для задания *url* LDAP.
  - ***revocation-check*** *method1 [method2]*
    - *method1* – параметр, принимающий одно из двух значений:
      crl – при проверке сертификата обязателен действующий CRL. Если действующий CRL не найден в базе продукта и его не удалось получить по протоколу LDAP, то сертификат не принимается
      none – при проверке сертификата действующий CRL используется, если он предустановлен в базе продукта или получен в процессе IKE обмена. Если это не так, то попытка получить CRL по протоколу LDAP не предпринимается и сертификат принимается.
    - *method2* – параметр необязательный, имеет одно значение:
      none – если действующий CRL не найден в базе продукта и его не удалось получить по протоколу LDAP, то сертификат принимается. Используется только тогда, когда *method1= crl*.
- ***crypto pki certificate chain*** *name* – для входа в режим настройки цепочки сертификатов CA:
  - *certificate* – для добавления CA сертификата (в шестнадцатеричном представлении) в базу Продукта:
  - можно задать несколько таких команд для добавления либо промежуточных CA сертификатов, либо любых CA сертификатов.

В отличие от Cisco наш Продукт не проверяет являются ли добавляемые сертификаты из одной цепочки. Поэтому, можно добавлять в один *trustpoint* не только промежуточные CA сертификаты, но вообще любые CA сертификаты.

При добавлении CA сертификата в *trustpoint* командой *crypto pki certificate chain* он автоматически добавляется в базу Продукта.

При старте cs_console при поиске сертификата проверяются все существующие *trustpoint's* в базе Продукта. В случае отсутствия соответствующего CA сертификата в базе Продукта, *trustpoint* автоматически удаляется из cisco-like конфигурации и, следовательно, удаляются все CA сертификаты, зарегистрированные в этом *trustpoint*. При этом выдается соответствующее сообщение в лог.

Создание ключевой пары и запроса на локальный сертификат

Создать ключевую пару и запрос на локальный сертификат для шлюза безопасности можно двумя путями:

- локально с помощью утилиты ***/opt/VPNagent/bin/cert_mgr create***
- на отдельном компьютере с помощью средств MS Windows и СКЗИ, как описано в этом документе.

Регистрация локального сертификата

Для регистрации локального сертификата в базе Продукта используется утилита командной строки *cert_mgr import*.

Удаление сертификатов

Удалять сертификаты из базы Продукта можно двумя способами:

- с помощью утилиты командной строки ***cert_mgr remove***
- через cs_console командой ***no crypto pki trustpoint***.

При удалении *trustpoint* с указанным именем, все CA сертификаты из этого *trustpoint* удаляются из текущей конфигурации, базы Продукта и cisco-like конфигурации.

Если в cs_console добавить сертификат в *trustpoint,* а потом, выйдя из консоли, удалить добавленный сертификат с помощью *cert_mgr remove*, то при следующем старте консоли *trustpoint* с сертификатом удалится и оттуда.

Удалить CRL из базы Продукта помощью утилиты командной строки ***cert_mgr remove*** невозможно. Если в команде указать номер (индекс) CRL, то будет выведено сообщение об ошибке о недопустимом индексе.

Просмотр сертификатов в базе Продукта

Для просмотра сертификатов в базе Продукта используйте команду ***cert_mgr show***.

Отсылка локального сертификата

Для отсылки локального сертификата партнеру по протоколу IKE:

в LSP-конфигурации (конфигурационный файл):

Для отсылки локального сертификата партнеру по протоколу IKE в LSP, в структуре ***AuthMethodGOSTSign*** задать атрибут ***SendCertMode*** со значением:

- *ALWAYS* – всегда отсылать локальный сертификат
- *CHAIN* – всегда отсылать локальный сертификат, CA сертификат и промежуточные CA сертификаты.

в cisco-like конфигурации (в интерфейсе командной строки):

при создании политики IKE, параметры которой согласовываются с партнером, в режиме команды ***crypto isakmp policy*** задать метод аутентификации сторон с использованием сертификатов командой

authentication rsa-sig

В файле настроек конвертора cs_conv.ini параметру *send_cert* присвоено значение *ALWAYS*, и поэтому по умолчанию партнеру всегда будет отсылаться локальный сертификат по протоколу IKE.

Получение сертификата партнера

Сертификат партнера можно получить либо по протоколу IKE, либо по протоколу LDAP.

Сначала шлюз безопасности пытается получить сертификат партнера по IKE. Если партнер не прислал сертификат, а прислал свой идентификатор, то шлюз безопасности по этому идентификатору ищет сертификат партнера сначала в своей базе Продукта, если не нашел, то продолжает поиск на LDAP-сервере.

Получение сертификата партнера по IKE

Для получения сертификата партнера по по протоколу IKE нужно:

в LSP-конфигурации:

- в локальной конфигурации в структуре *AuthMethodGOSTSign* задать атрибут *SendRequestMode* со значением *ALWAYS* – всегда запрашивать сертификат партнера
- в конфигурации партнера в структуре *AuthMethodGOSTSign* задать атрибут *SendCertMode* со значением:
  - *ALWAYS* – высылать сертификат
  - *CHAIN* – высылать локальный сертификат, CA сертификат с цепочкой промежуточных CA.

в cisco-like конфигурации:

в режиме команды ***crypto isakmp policy*** задать метод аутентификации сторон с использованием сертификатов командой

authentication rsa-sig

В файле настроек конвертора ***cs_conv.ini*** параметру *send_request* присвоено значение *ALWAYS*, и поэтому по умолчанию у партнера всегда будет запрашиваться локальный сертификат по протоколу IKE.

Получение сертификата партнера по LDAP

Получение сертификата партнера на LDAP-сервере. В этом случае партнер присылает свой идентификатор, а шлюз безопасности по значению Subject будет искать сертификат партнера на LDAP-сервере. Для прохождения LDAP-пакетов до LDAP-сервера необходимо в политике задать соответствующий фильтр.

В LSP-конфигурации:

в локальной конфигурации задать структуру ***LDAPSettings*** с IP-адресом LDAP-сервера и также:

- если прислан идентификатор типа DN:
  - шлюз безопасности по Subject ищет сертификат партнера сначала в своей базе Продукта, а затем на LDAP-сервере
- если прислан идентификатор другого типа:
  - для получения Subject в локальной конфигурации задаются атрибуты *RemoteID, RemoteCredential, DoNotMapRemoteIDToCert*
  - если *DoNotMapRemoteIDToCert = TRUE*, то Subject будет составляться из *RemoteCredential*
  - если *DoNotMapRemoteIDToCert = FALSE*, то Subject будет составляться из *RemoteCredential и RemoteID*.
  - по составленному значению Subject шлюз безопасности ищет сертификат партнера сначала в своей базе Продукта, а затем на LDAP-сервере.

В cisco-like конфигурации:

если партнер не прислал свой сертификат по протоколу IKE, и в базе Продукта его нет, то шлюз безопасности посылает запрос на заданный LDAP-сервер в команде ***crl query*** для получения сертификата партнера. По полученному идентификатору типа *dn* от партнера будет осуществляться поиск сертификата. Если получен идентификатор другого типа – запрос на LDAP-сервер не посылается. Если отредактировать сконвертированную native-конфигурацию для работы с идентификаторами другого типа, как описано в предыдущем пункте, то сертификат партнера можно получить по LDAP.

Проверка сертификата по CRL

Для проверки сертификата партнера по списку отозванных сертификатов (CRL) нужно:

в LSP-конфигурации:

в структуре GlobalParameters задать атрибут ***CRLHandlingMode***, при значениях этого атрибута:

- *optional* – используется действующий CRL из базы Продукта
- *enable* и *best_effort* – действующий CRL может быть получен по LDAP.

Для получения CRL с LDAP-сервера сначала проверяется поле CDP в проверяемом сертификате, если поле CDP отсутствует, то в конфигурации должна быть задана структура ***LDAPSettings*** с адресом LDAP-сервера. В базу Продукта с LDAP-сервера загружается действующий CRL и по нему проверяется сертификат партнера.

Для прохождения LDAP-пакетов до LDAP-сервера необходимо в политике задать соответствующий фильтр.

в cisco-like конфигурации:

в режиме команды *crypto pki trustpoint* командой ***revocation-check*** задается режим использования CRL.

Несколько локальных и CA сертификатов

Иногда при работе с разными партнерами аутентификация осуществляется с использованием разных локальных сертификатов, подписанных разными УЦ, соответственно и CA сертификаты разные.

В cisco-like конфигурации:

в командной строке нет команд для указания соответствия между идентификатором партнера, локальным сертификатом и CA сертификатом. Поэтому после конвертирования cisco-like конфигурации в LSP конфигурацию последнюю необходимо отредактировать.

В LSP-конфигурации:

в структуре *AuthMethodGOSTSign* существуют атрибуты, которые позволяют задать соответствие между локальным, партнерским и CA сертификатами, локальным и партнерским идентификаторами.

Пример работы с сертификатами, выпущенными разными УЦ, опубликован на сайте компании http://www.s-terra.com/ RU/support/support.htm в разделе «Типовые сценарии» – «Построение VPN туннеля в топологии «звезда» с использованием сертификатов, выпущенных разными УЦ».

Расширения сертификата (Certificate Extensions)

Имеются некоторые ограничения при работе с расширениями сертификата (Extensions), которые помечены как критичные. В таблице приведен список расширений сертификата, которые будут распознаваться и обрабатываться Продуктом, если у них установлен признак критичности TRUE. Если в сертификате будут присутствовать другие расширения, не указанные в таблице и заданные как критичные, то такой сертификат не может быть использован. Если же расширение отсутствует в таблице, но является некритичным, то оно игнорируется, и сертификат используется.

Name	OID value
Subject Key Identifier	2.5.29.14
Key Usage	2.5.29.15
Subject Alternative Name	2.5.29.17
Issuer Alternative Name	2.5.29.18
Basic Constraints	2.5.29.19
Name Constraints	2.5.29.30
CRL Distribution Points	2.5.29.31
Authority Key Identifier	2.5.29.35

Описания значений и полный список Certificate Extensions можно посмотреть в документе RFC 5280 (http://tools.ietf.org/html/rfc5280#section-4.2).

Создание локального сертификата с использованием СКЗИ "КриптоПро CSP"

Для создания локального сертификата нужно создать ключевую пару и запрос на локальный сертификат. Это можно сделать, используя алгоритмы ГОСТ, средствами Microsoft Windows.

В качестве внешней криптографической библиотеки используется СКЗИ "КриптоПро CSP". Установка и настройка СКЗИ "КриптоПро CSP" описаны в этой главе.

Установка СКЗИ "КриптоПро CSP"

На отдельном компьютере с ОС Microsoft Windows установите СКЗИ "КриптоПро CSP 3.6".

При инсталляции выбирайте: вид установки – Выборочная.

Компоненты, которые необходимо установить:

- Криптопровайдер уровня ядра ОС
- Совместимость с КриптоПро CSP 3.0.

Настройка СКЗИ "КриптоПро CSP"

При аутентификации сторон при помощи сертификатов, необходимо провести некоторые настройки в СКЗИ "КриптоПро CSP".

Для хранения секретного ключа локального сертификата используется контейнер, который может быть защищен паролем. Контейнер размещается:

- либо на внешнем ключевом носителе, который должен храниться только у администратора
- либо на локальном ключевом носителе (Реестр) на компьютере администратора.

СКЗИ "КриптоПро CSP" умеет считывать секретный ключ из контейнера как на внешнем ключевом носителе. так и на локальном ключевом носителе.

Локальный ключевой считыватель

Если контейнер с секретным ключом локального сертификата надо разместить в Реестре, то его нужно инсталлировать как считыватель. Такая инсталляция описана в разделе “Инсталляция ключевого считывателя Реестр в "КриптоПро CSP 3.6".

Внешний ключевой считыватель и носитель информации

Если контейнер будет расположен на внешнем ключевом носителе, то сначала нужно подключить к компьютеру считыватель ключевой информации, а затем инсталлировать его. Подключение внешних считывателей ключевой информации описано в разделе "Подключение внешних ключевых считывателей".

После установки "КриптоПро CSP 3.6" сразу же инсталлированы все считыватели смарт-карт и все съемные диски, а остальные считыватели нужно инсталлировать. Для eToken и дисковода инсталляция считывателя уже выполнена.

Для некоторых внешних считывателей еще нужно выполнить Инсталляцию носителей. После установки "КриптоПро CSP 3.6" сразу же инсталлированы несколько типов носителей для eToken, остальные нужно инсталлировать.

Процедура инсталляции внешних считывателей и носителей описана в разделе "Инсталляция внешнего считывателя и ключевого носителя информации в "КриптоПро CSP 3.6".

Подключение внешних ключевых считывателей (носителей)

Подключите внешний ключевой считыватель к компьютеру, следуя прилагаемой инструкции (Не следует подключать eToken до установки драйверов). Установите все необходимые файлы и драйверы для работы внешнего считывателя, прилагаемые к нему.

В состав дистрибутива СКЗИ "КриптоПро CSP 3.6" не входят драйвера, обеспечивающие взаимодействие "КриптоПро CSP 3.6" с внешними ключевыми носителями.

Для этого с Web-страницы компании Крипто-ПРО http://www.cryptopro.ru/CryptoPro/products/csp/readers.htm загрузите и установите модуль поддержки внешнего считывателя для СКЗИ "КриптоПро CSP 3.6".

Создание локального сертификата в “КриптоПро CSP 3.6”

Инсталляция ключевого считывателя Реестр в "КриптоПро CSP 3.6"

Для инсталляции локального ключевого считывателя Реестр надо выполнить следующие действия:

Шаг 1: запустить КриптоПро CSP: `Пуск – Настройка – Панель управления – КриптоПро CSP`

Шаг 2: в появившемся окне Свойства войти во вкладку Оборудование и нажать кнопку `Настроить считыватели...` (Рисунок 1):

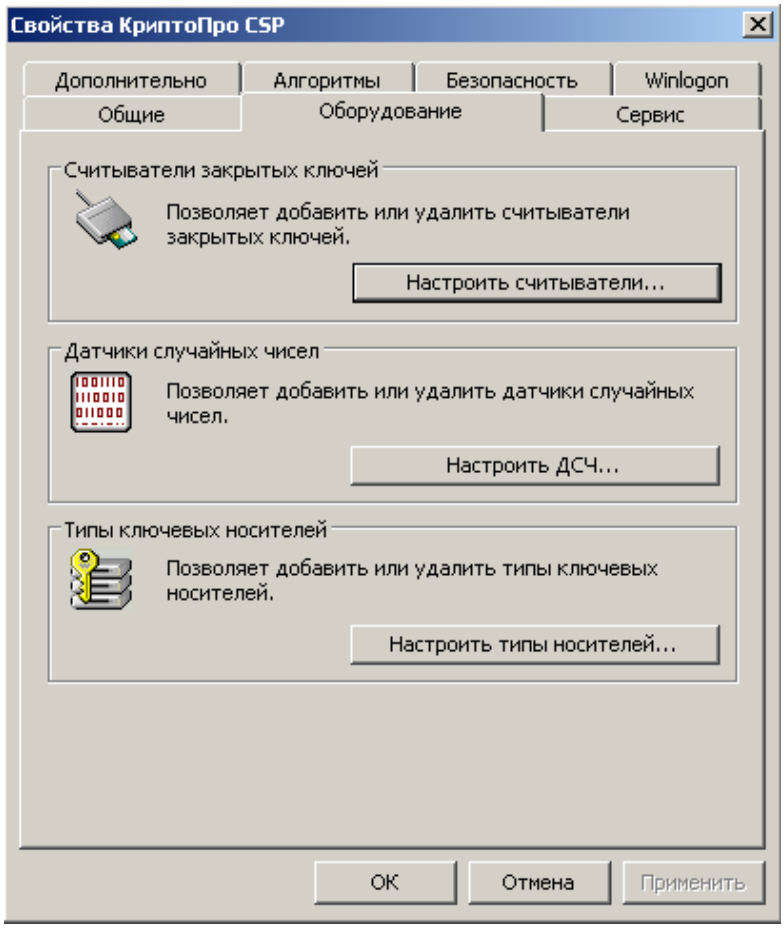

Рисунок 1

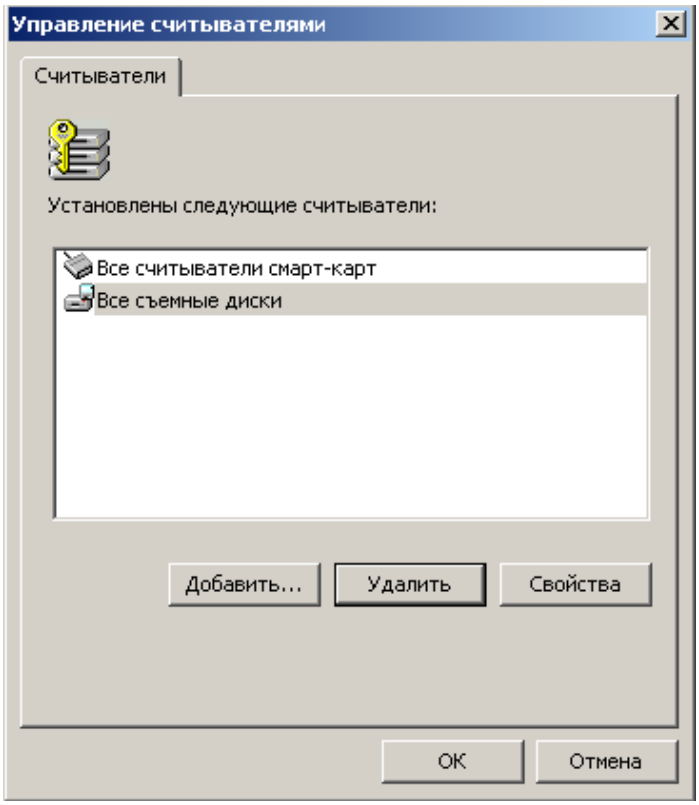

Рисунок 2

Шаг 3: нажать кнопку `Добавить...`, чтобы добавить новый ключевой носитель (Рисунок 2).

Шаг 4: в окне визарда для инсталляции считывателя нажать кнопку `Далее`:

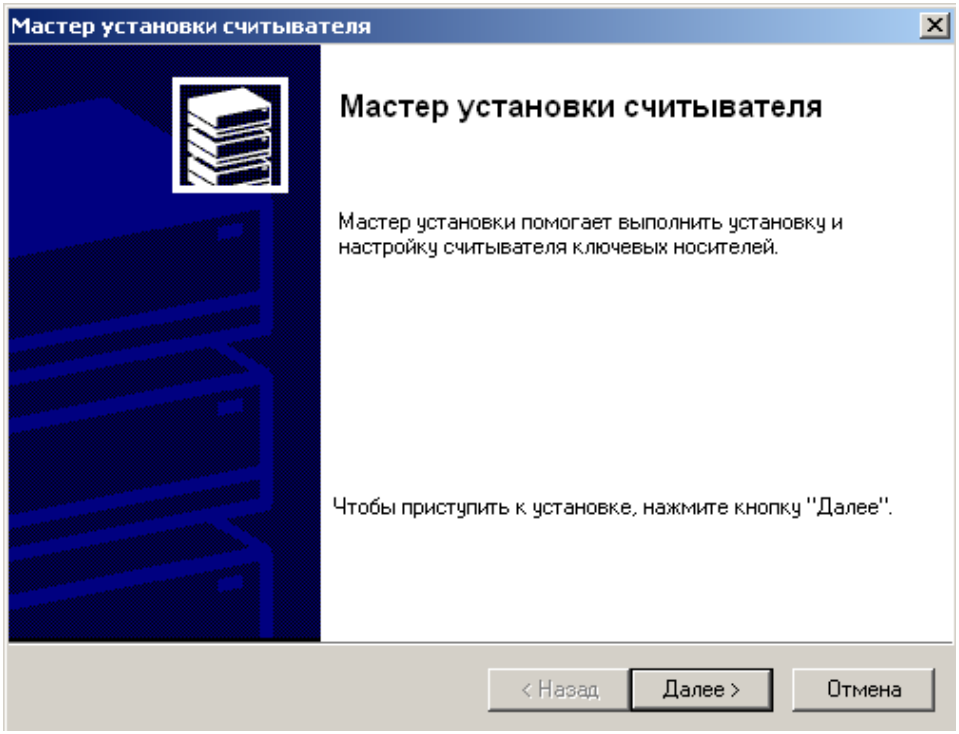

Рисунок 3

Шаг 5: из представленного списка выбрать считыватель "Реестр" и нажать кнопку `Далее`:

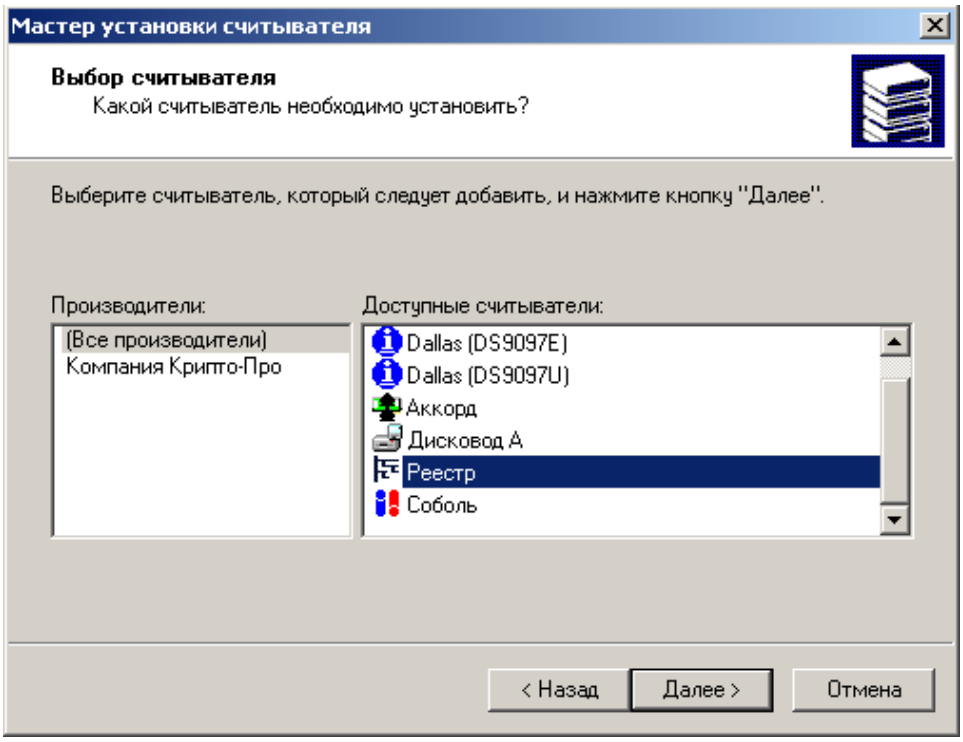

Рисунок 4

Шаг 6: считывателю Реестр можно присвоить имя и нажать кнопку `Далее`:

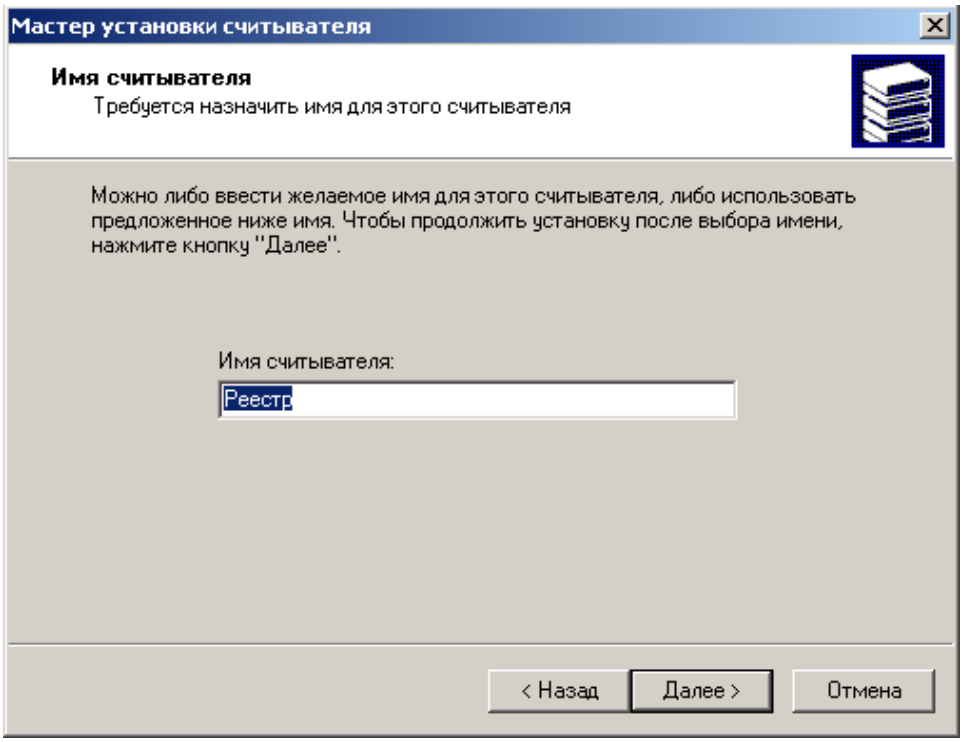

Рисунок 5

Шаг 7: инсталляция считывателя Реестр завершена, нажмите `Готово`:

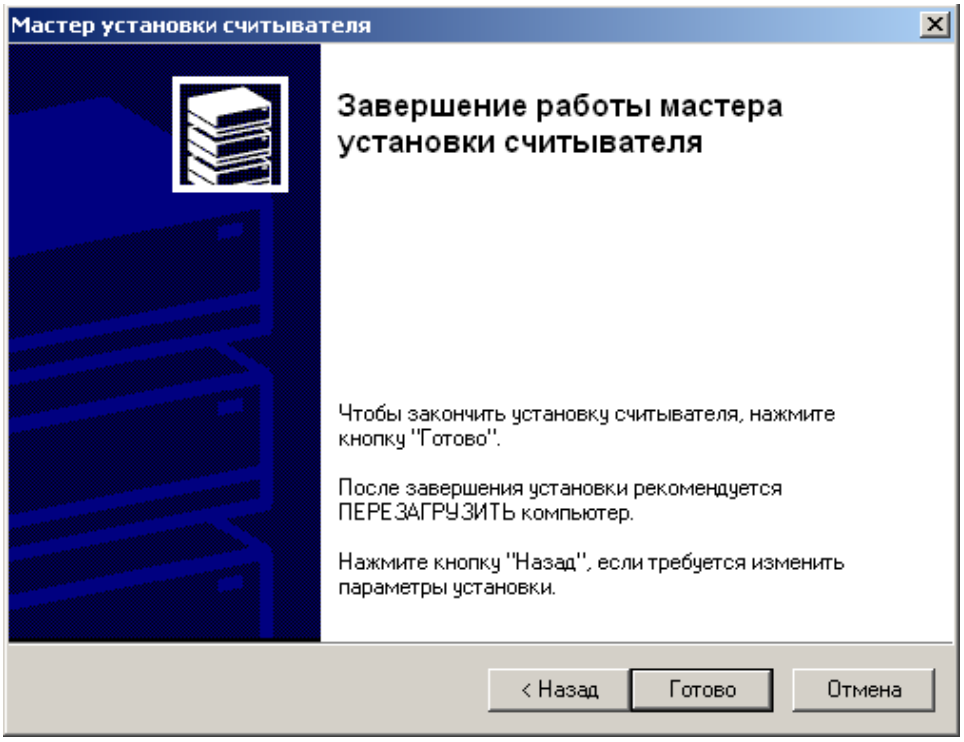

Рисунок 6

Шаг 8: считыватель Реестр добавлен в список инсталлированных считывателей, нажмите `OK`:

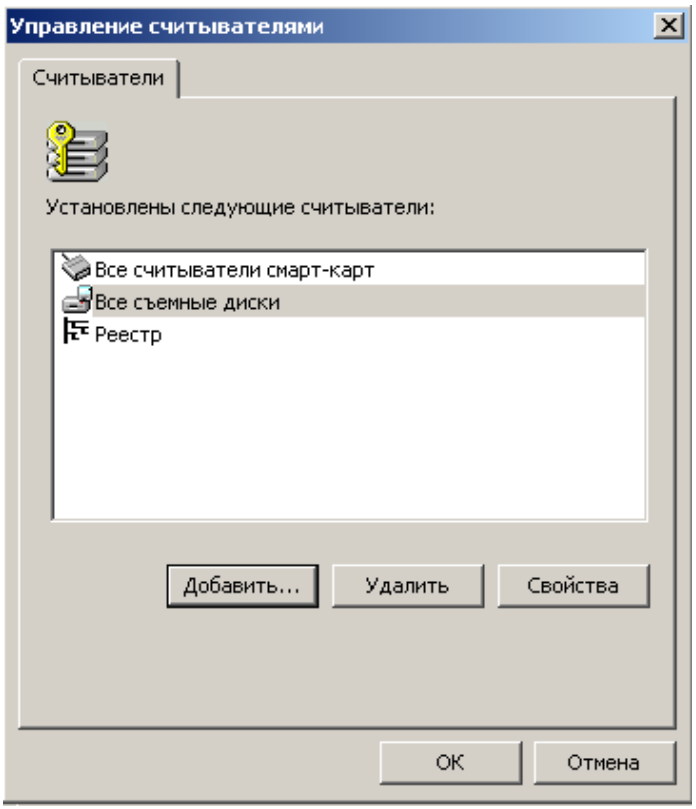

Рисунок 7

Шаг 9: перезагрузите компьютер.

Инсталляция внешнего считывателя и ключевого носителя информации в "КриптоПро CSP 3.6"

Инсталляция некоторых внешних считывателей уже выполнена. Для остальных внешних считывателей инсталляция выполняется так же как и для Реестра, описанная в разделе "Инсталляция ключевого считывателя Реестр в "КриптоПро CSP 3.6".

Для некоторых считывателей необходимо еще выполнить инсталляцию носителей. Для Rutoken и eToken Pro такая инсталляция уже выполнена.

Шаг 1: запустить КриптоПро CSP: `Пуск - Настройка - Панель управления - КриптоПро CSP`

Шаг 2: Инсталляция других носителей производится нажатием клавиши `Настроить типы носителей...` в окне Свойства КриптоПро CSP во вкладке Оборудование:

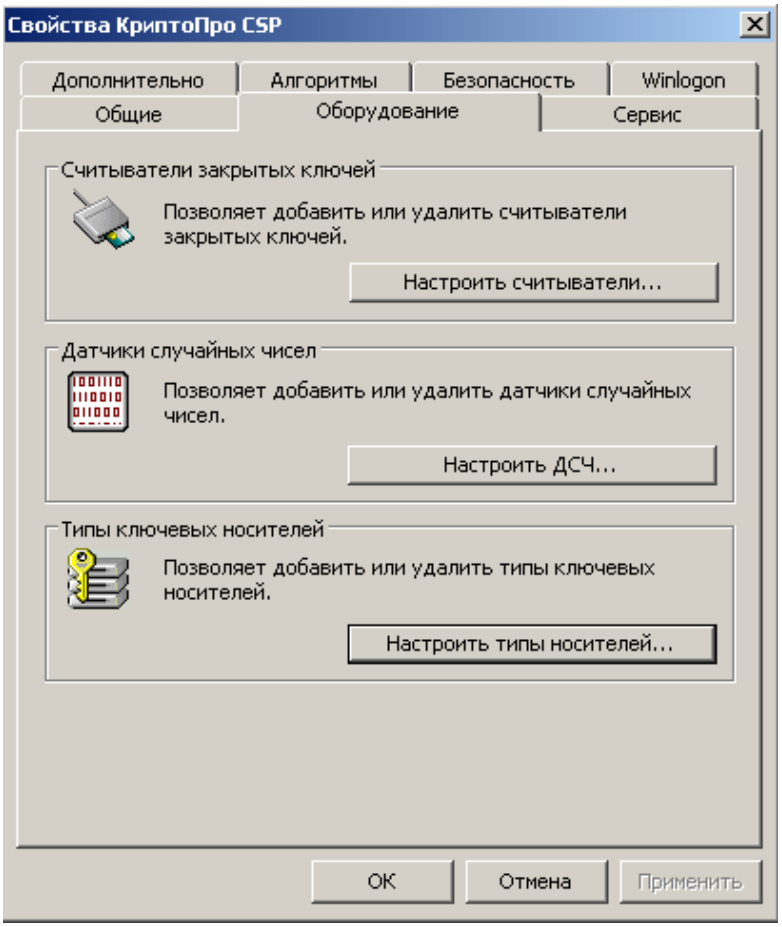

Рисунок 8

Шаг 3: вкладка Ключевые носители показывает инсталлированные ключевые носители. Для добавления носителя нажмите кнопку `Добавить...`

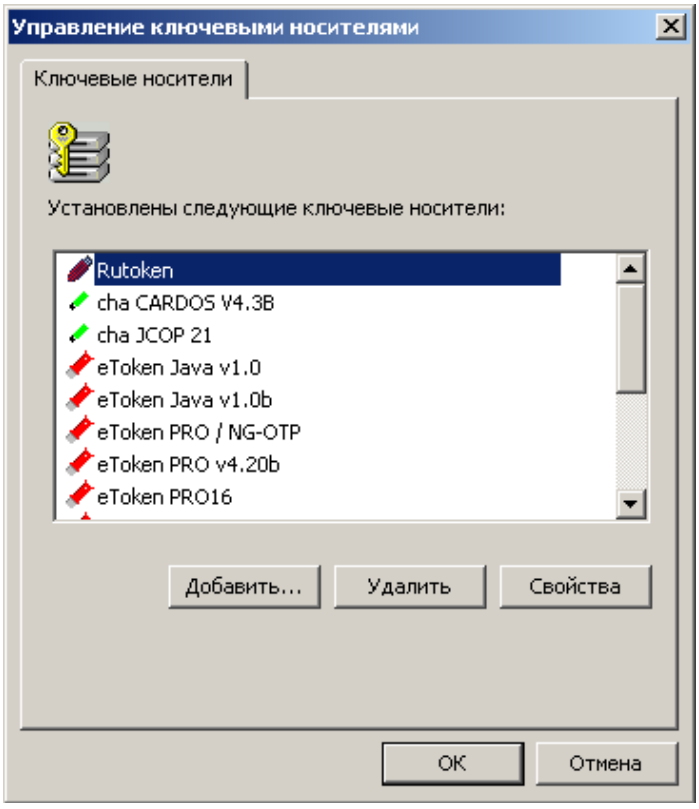

Рисунок 9

Далее следуйте указаниям Мастера установки ключевого носителя.

По завершению инсталляция внешнего ключевого носителя и считывателя полностью выполнена.

Установка и настройка Удостоверяющего Центра. Создание CA сертификата

Перед созданием ключевой пары и создания запроса на локальный сертификат опишем, как создать Удостоверяющий Центр (Центр Сертификации – CA) средствами MS, который будет выдавать локальный сертификат для шлюза безопасности. Если Вам известен Сертификационный Центр, который по Вашему запросу будет издавать сертификат, то перейдите к следующему разделу – созданию ключевой пары, в противном случае – создайте свой Удостоверяющий Центр.

На отдельном компьютере установите ОС Windows 2003 Server (SP2) и СКЗИ «КриптоПро CSP 3.6». Сервис Internet Information Services (IIS) должен быть включен.

Также инсталлируйте ключевой носитель, например Реестр, для хранения контейнера с секретным ключом CA сертификата.

Для инсталляции Удостоверяющего Центра Microsoft Certification Authority выполните следующие шаги:

Шаг 1: в окне установки компонент Windows (`Start-Settings-Control Panel-Add/Remove Programs-Add/Remove Windows Components`) установите флажок `Application Server` (Рисунок 10) и нажмите кнопку Details.

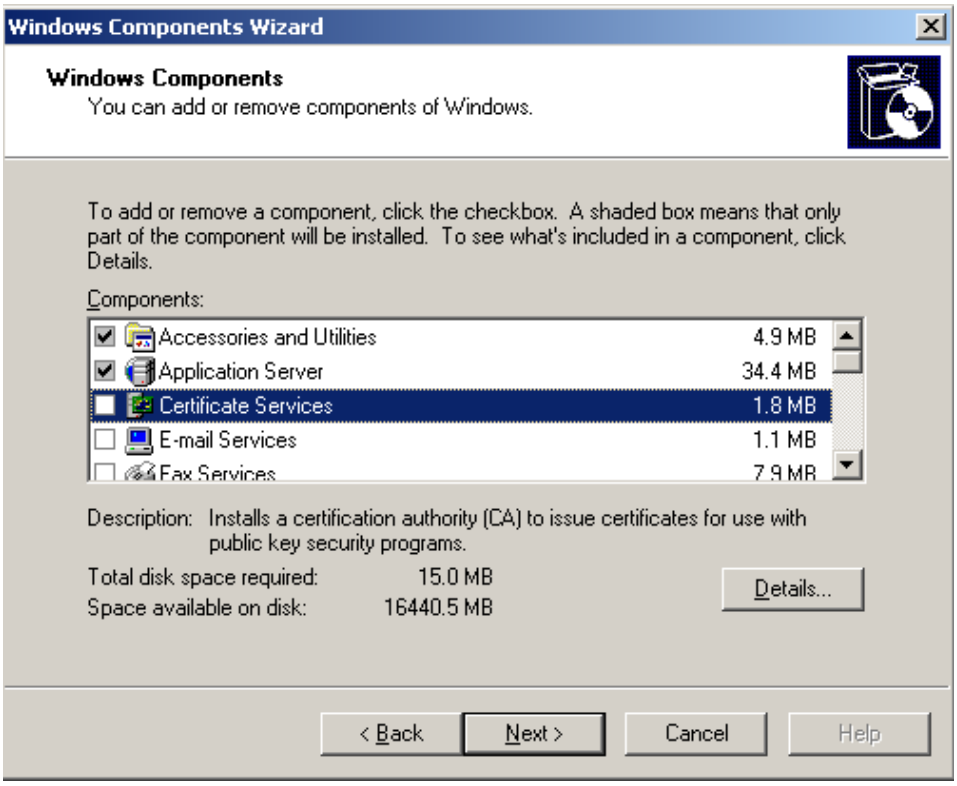

Рисунок 10

Установите сервисы Internet Information Services (IIS) и ASP.NET, и нажмите ОК (Рисунок 11).

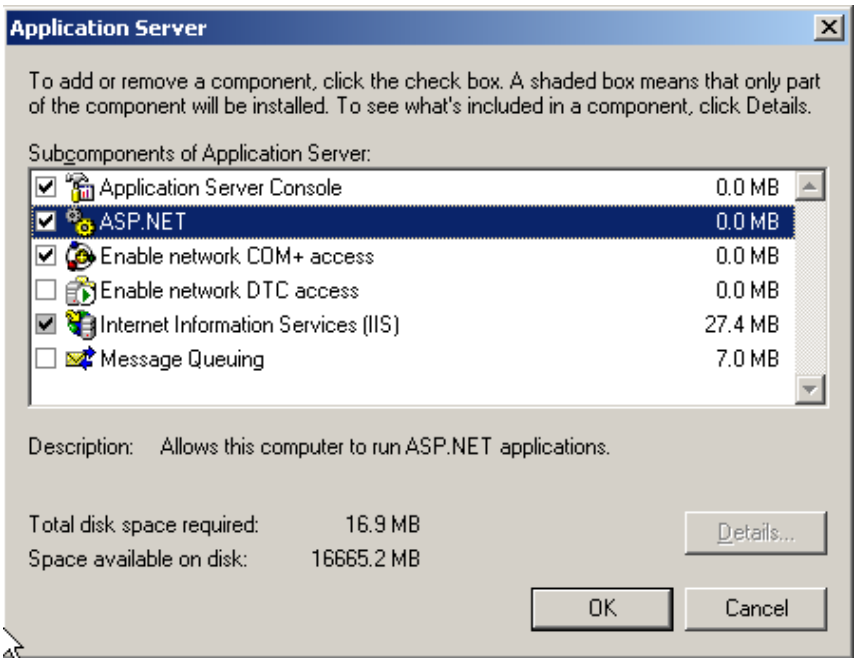

Рисунок 11

Шаг 2: установите сертификатный сервис: в окне установки компонент Windows установите флажок `Certificate Services` и нажмите `Next`:

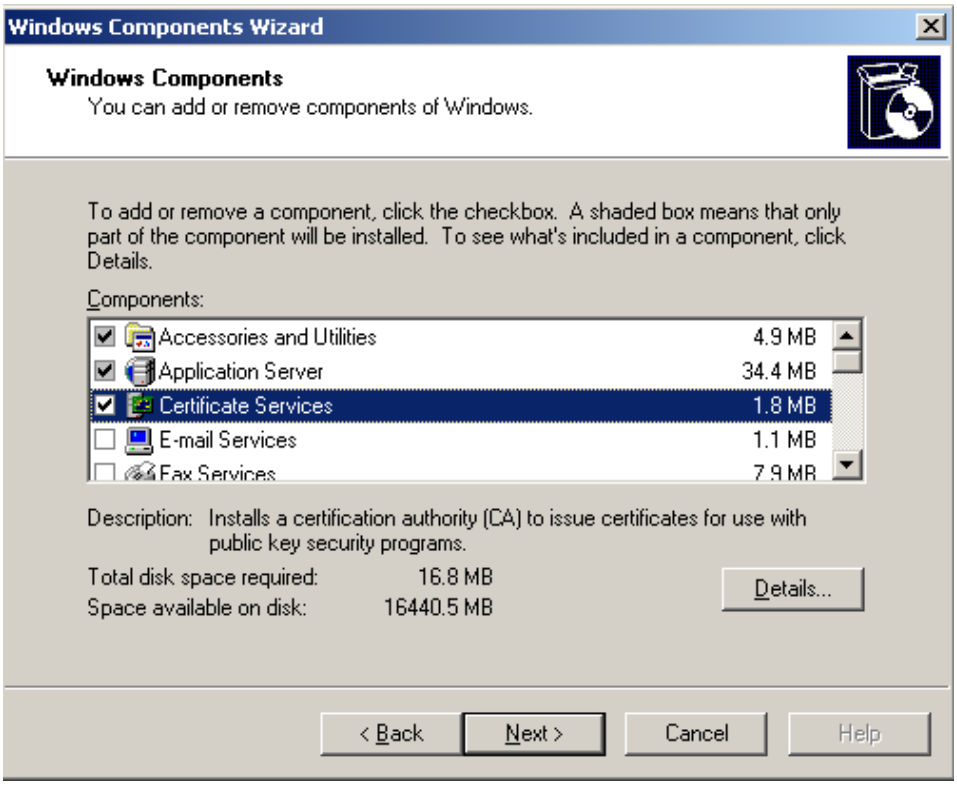

Рисунок 12

Если `Certificate Services` уже установлен, то его нужно удалить (снять флажок `Certificate Services`), а потом снова установить.

После установки флажка Certificate Services буде выдано предупреждение (Рисунок 13), нажмите кнопку Yes.

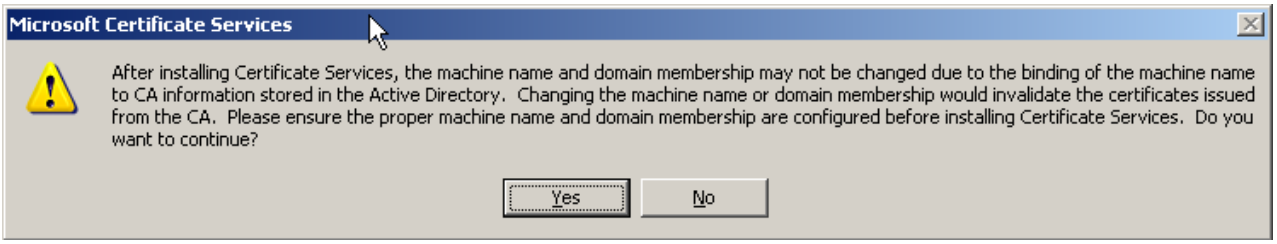

Рисунок 13

Шаг 3: в следующем окне выберите Удостоверяющий Центр с корневым CA сертификатом: поставьте переключатель в положение Stand-alone root CA. Установите флажок Use custom settings to generate the key pair and CA certificate и нажмите Next:

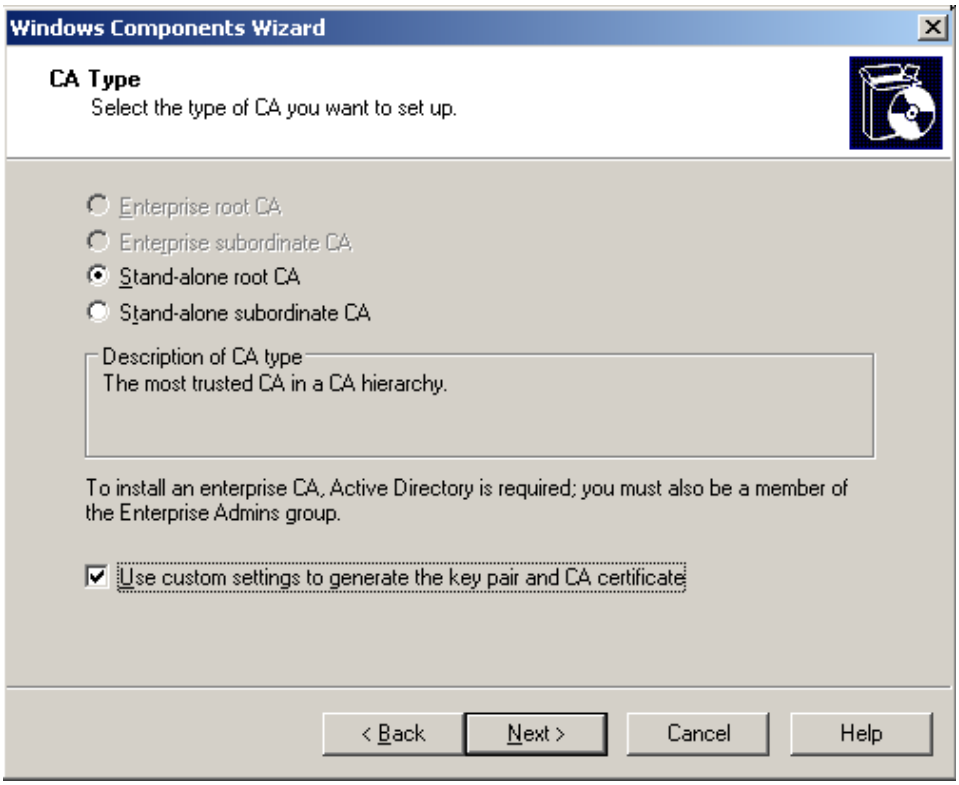

Рисунок 14

Шаг 4: в качестве криптопровайдера выберите Crypto-Pro GOST R34.10-2001 KC1 CSP, а в качестве хэш-алгоритма - GOST R34.11-94 и нажмите `Next`:

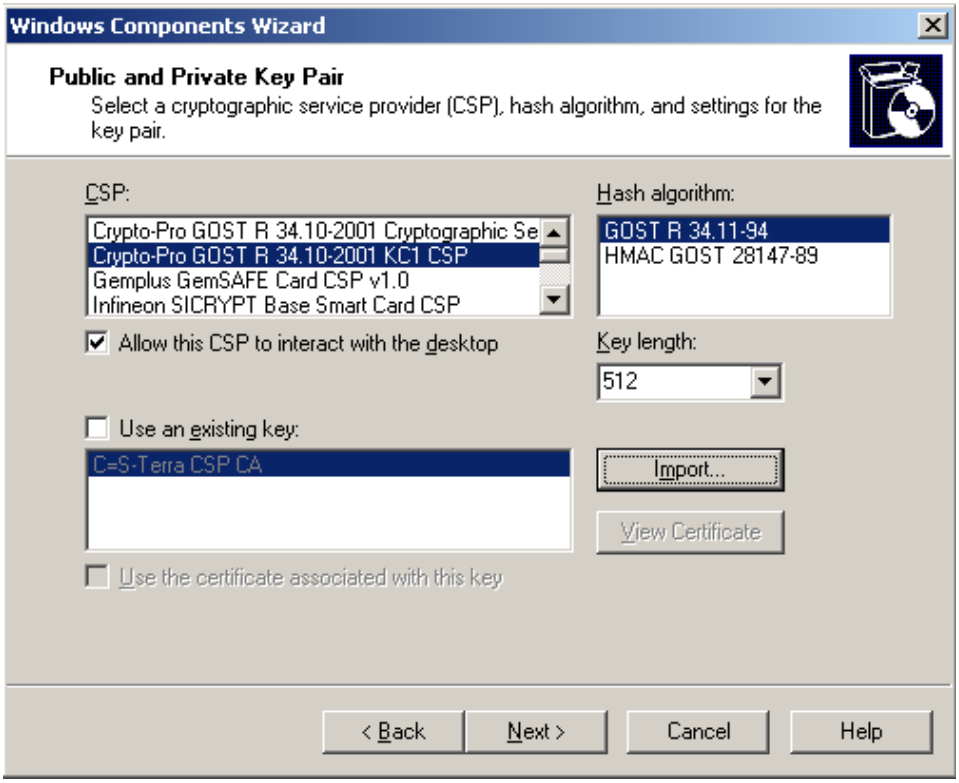

Рисунок 15

Шаг 5: заполните поля для CA сертификата и нажмите `Next`:

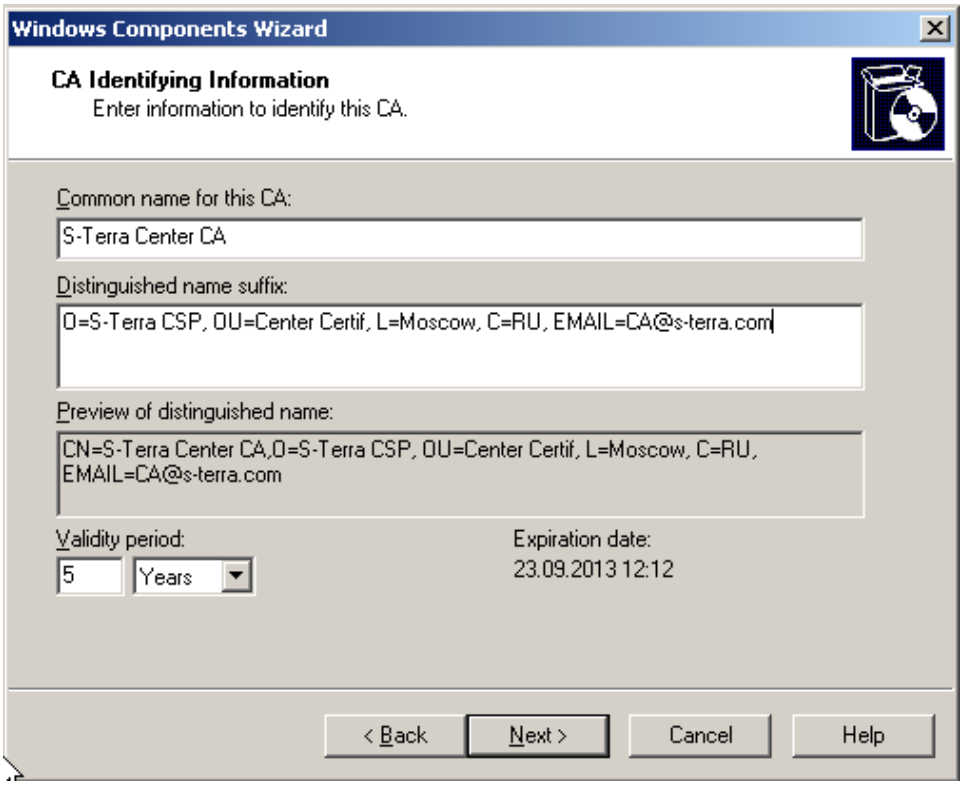

Рисунок 16

Шаг 6: выберите ключевой носитель Реестр, на котором будет записан контейнер с секретным ключом для CA сертификата и нажмите `OK`:

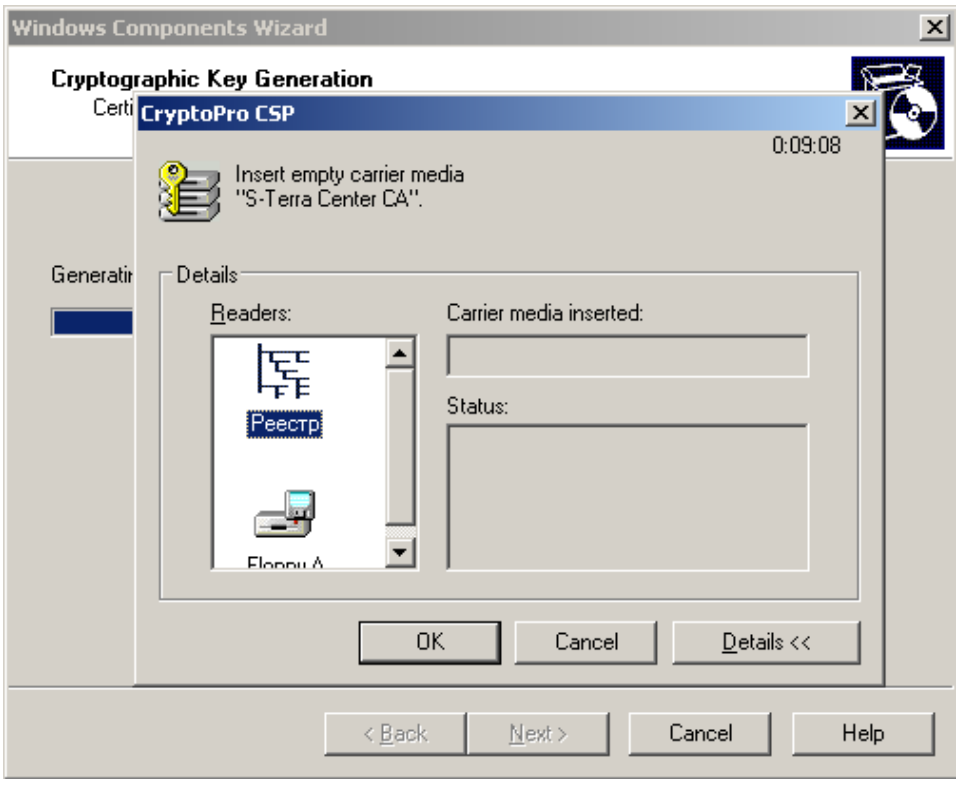

Рисунок 17

Шаг 7: подвигайте мышкой или нажмите любую клавишу пока происходит создание ключевой пары для CA сертификата:

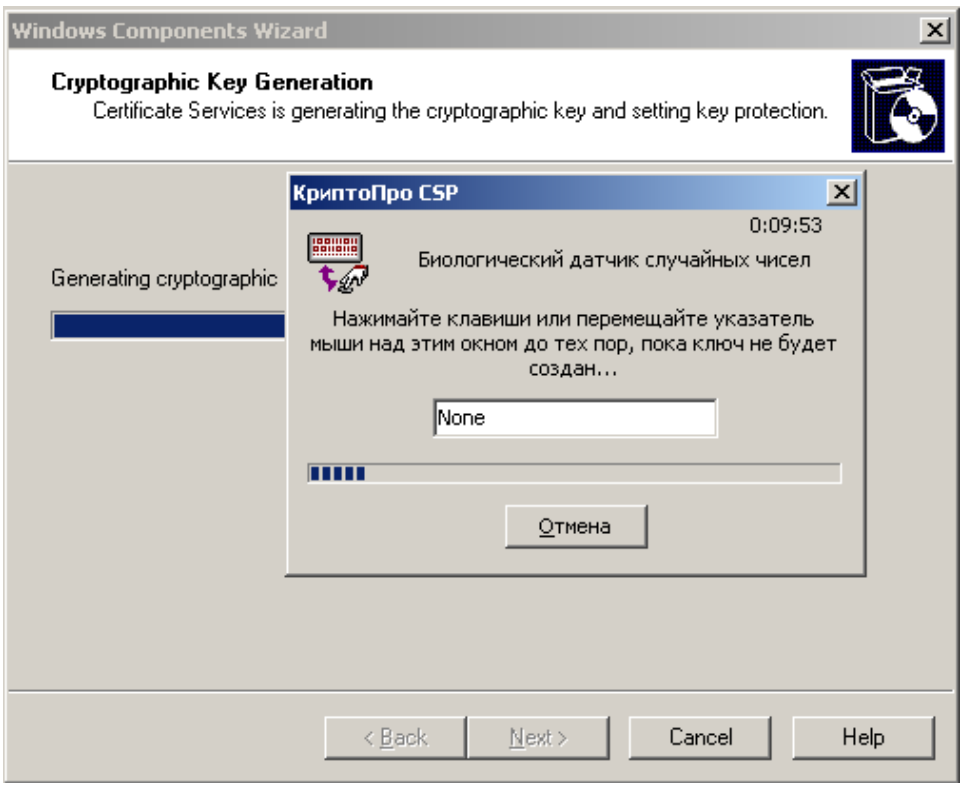

Рисунок 18

Шаг 8: задайте пароль к ключевому контейнеру и нажмите `OK`:

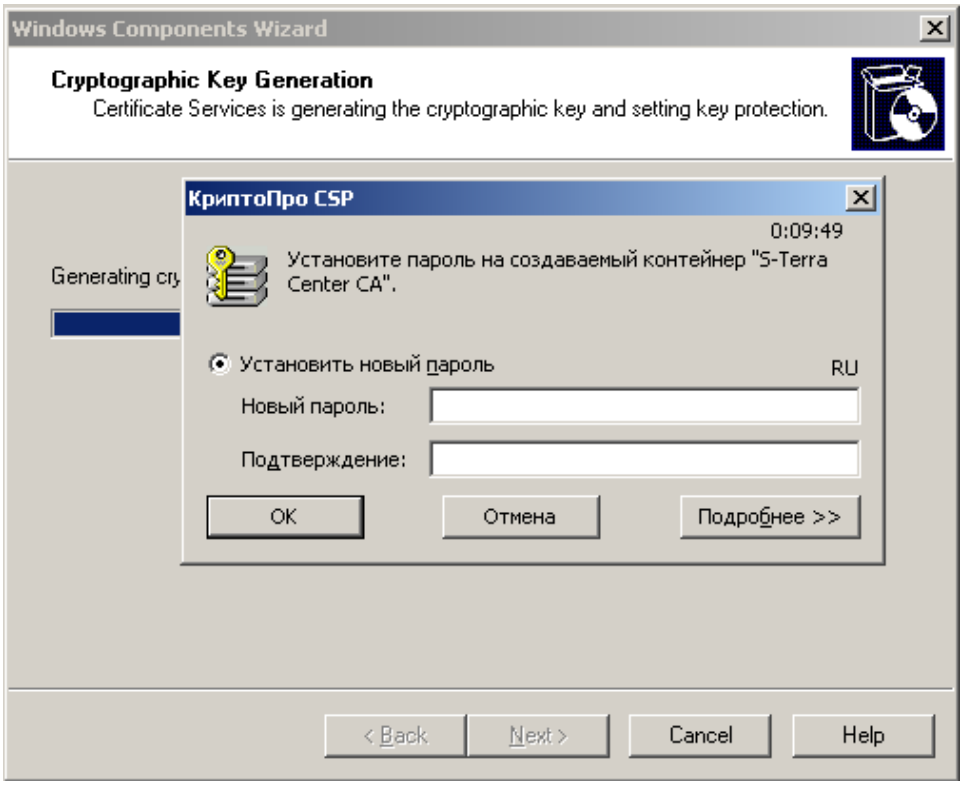

Рисунок 19

Шаг 9: в окне с указанием о размещении хранилища оставьте значения по умолчанию и нажмите `Next`:

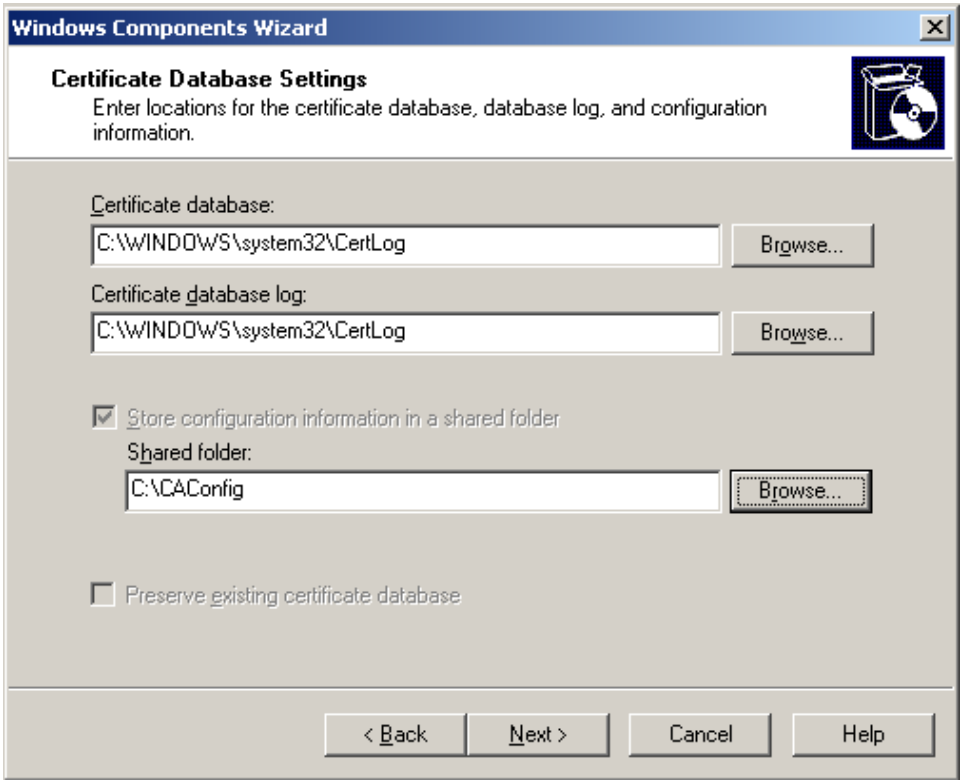

Рисунок 20

Шаг 10: во время настройки компонент Windows выдается следующий запрос на включение компоненты Active Server Pages (Рисунок 21), нажмите кнопку `Yes.`:

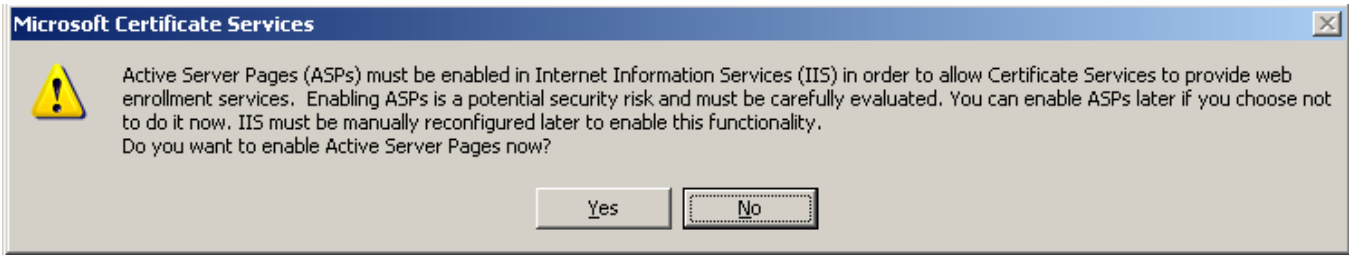

Рисунок 21

Шаг 11: инсталляция Удостоверяющего Центра завершена, нажмите `Finish.`

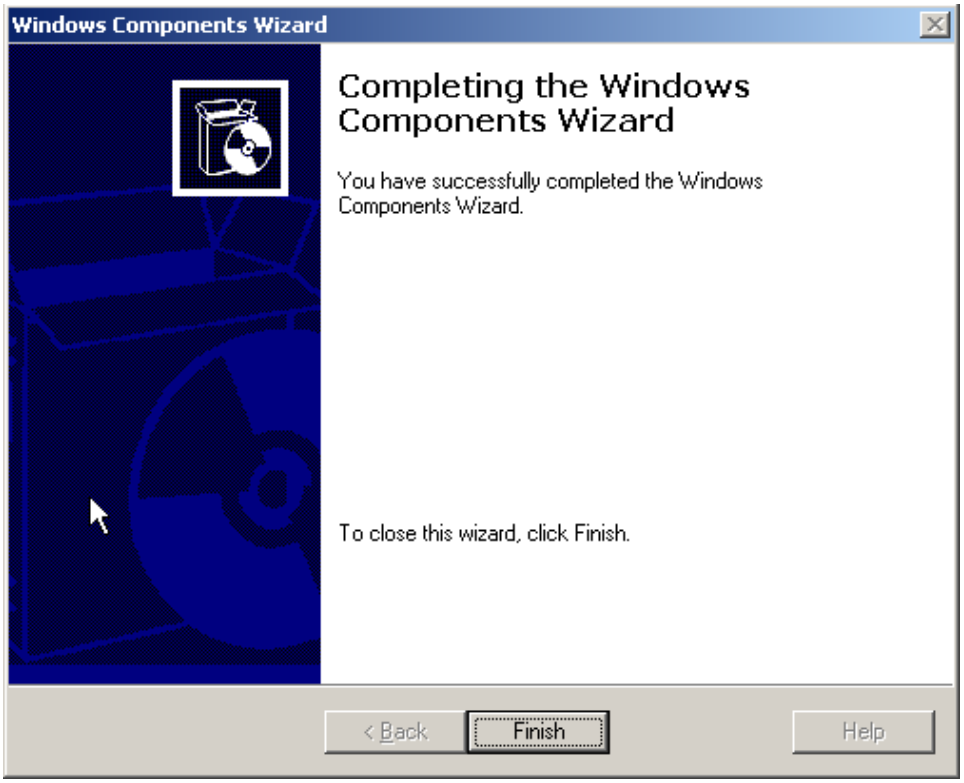

Рисунок 22

Шаг 12: для экспортирования CA сертификата в файл, войдите сначала в Certificate Authority `(Start-Settings-Control Panel-Administrative Tools-Certificate Authority)`, выделите центр CA и нажмите `Properties:`

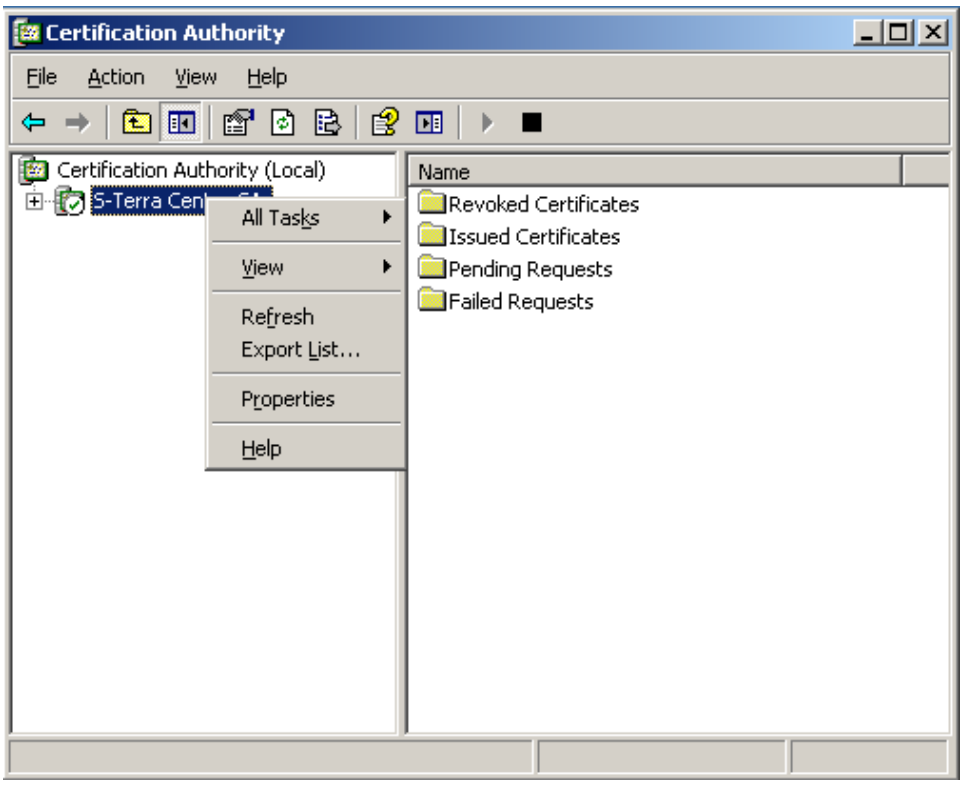

Рисунок 23

Шаг 13: далее во вкладке General нажмите кнопку `View Certificate`. В появившемся окне Certificate выберите вкладку Details, в выпадающем меню `Show` выберите предложение All, чтобы увидеть все поля сертификата. Нажмите кнопку `Copy to File`:

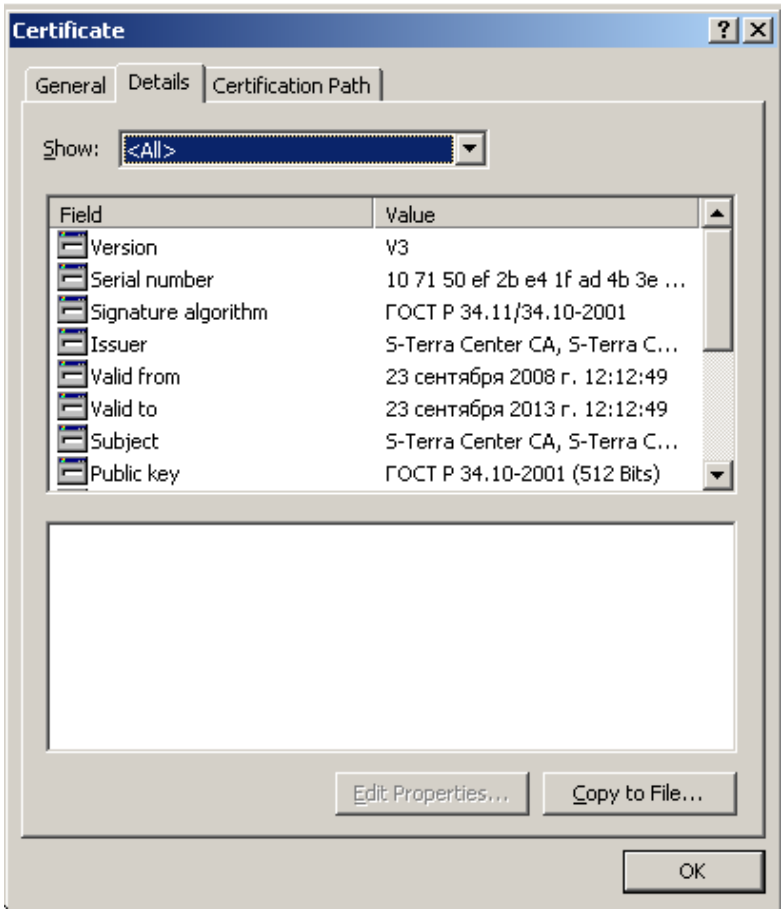

Рисунок 24

Шаг 14: в окне визарда экспортирования сертификата нажмите `Next`:

Рисунок 25

Шаг 15: выберите формат для сертификата – установите переключатель в первое положение:

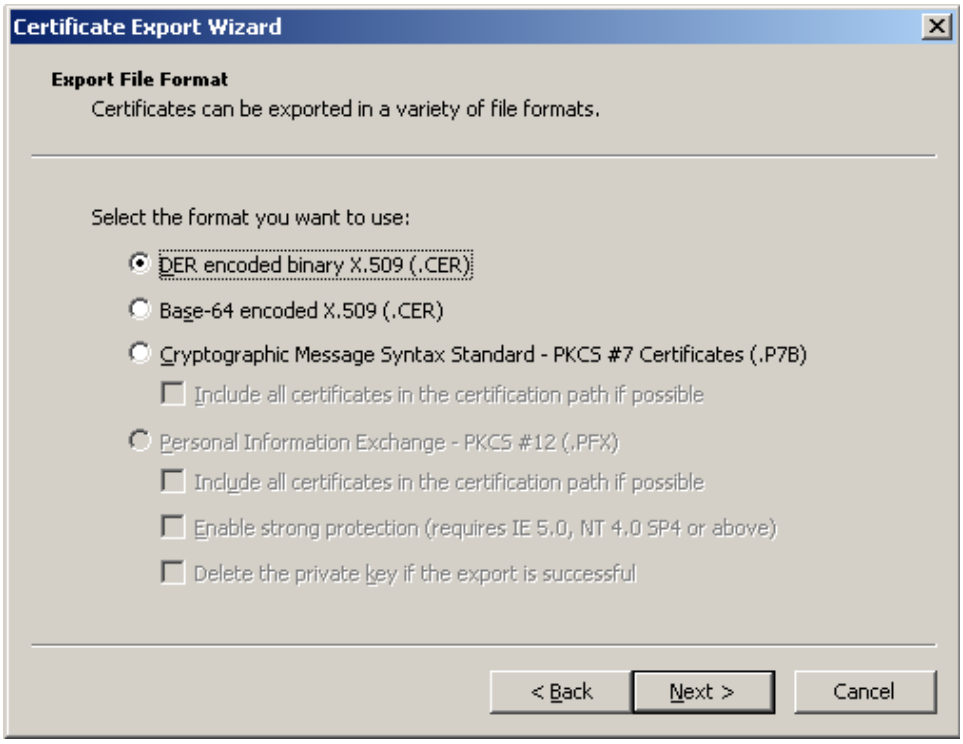

Рисунок 26

Шаг 16: выберите для сертификата имя файла, в который он будет экспортирован:

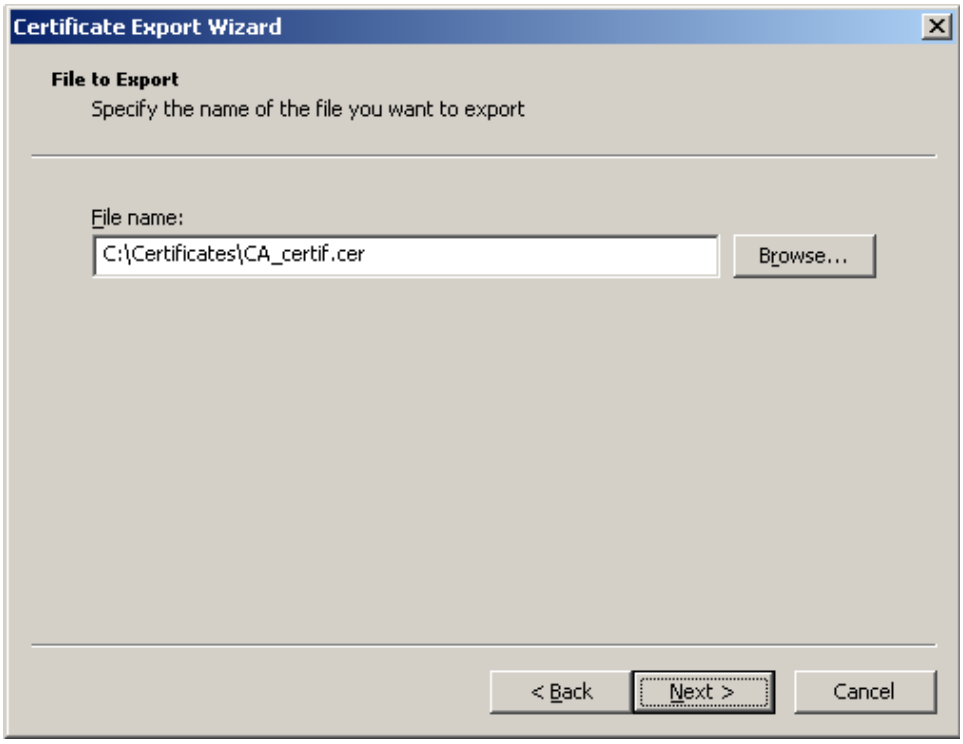

Рисунок 27

Шаг 17: экспортирование CA сертификата в файл завершено, нажмите `Finish`:

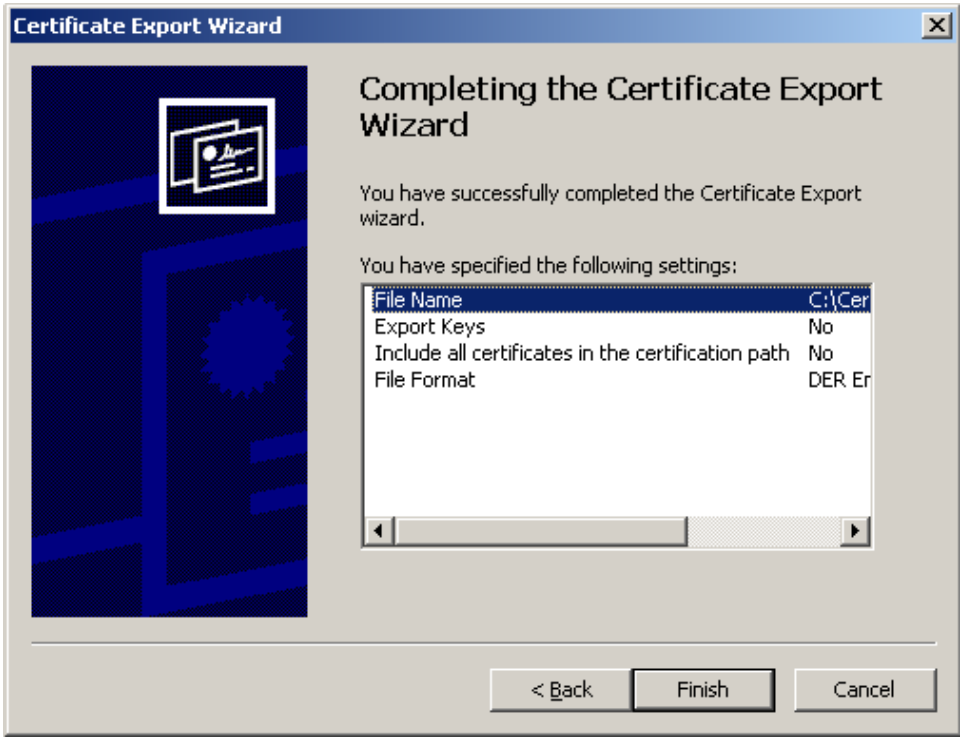

Рисунок 28

Закройте окно Cerificate, нажав кнопку OK (Рисунок 24).

Шаг 18: для автоматического создания подписываемых сертификатов по запросу проведите некоторые настройки Удостоверяющего Центра. В окне Properties войдите во вкладку Policy Module (Рисунок 29) и нажмите кнопку `Properties...`

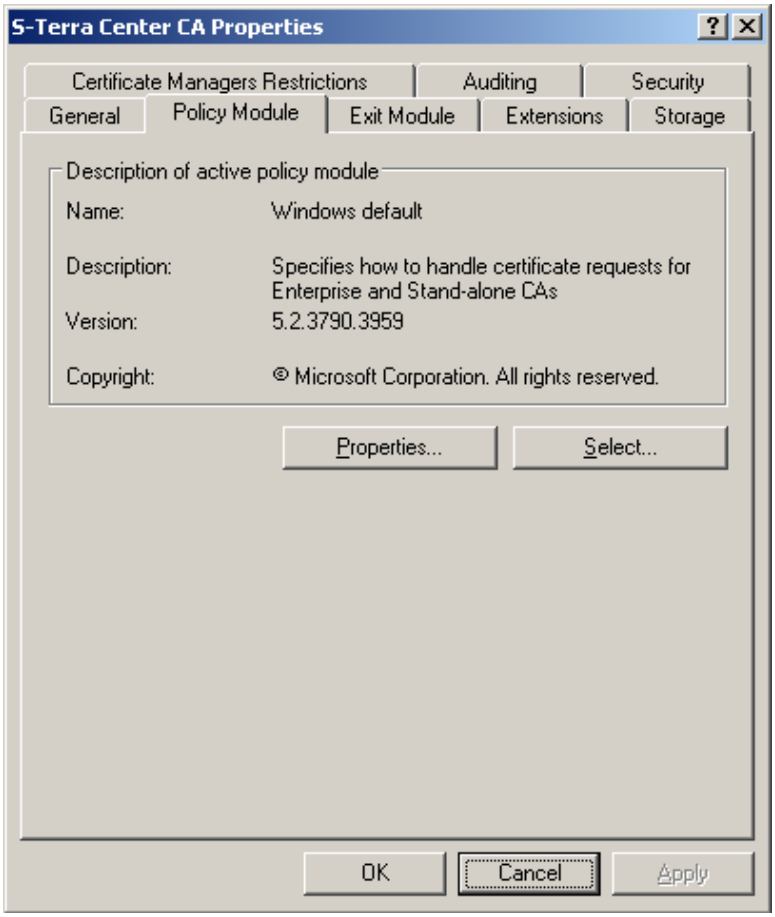

Рисунок 29

Шаг 19: в появившемся окне Properties (Рисунок 30) установите переключатель в положение `Follow the settings …` (автоматически издавать сертификат по запросу) и нажмите `OK`:

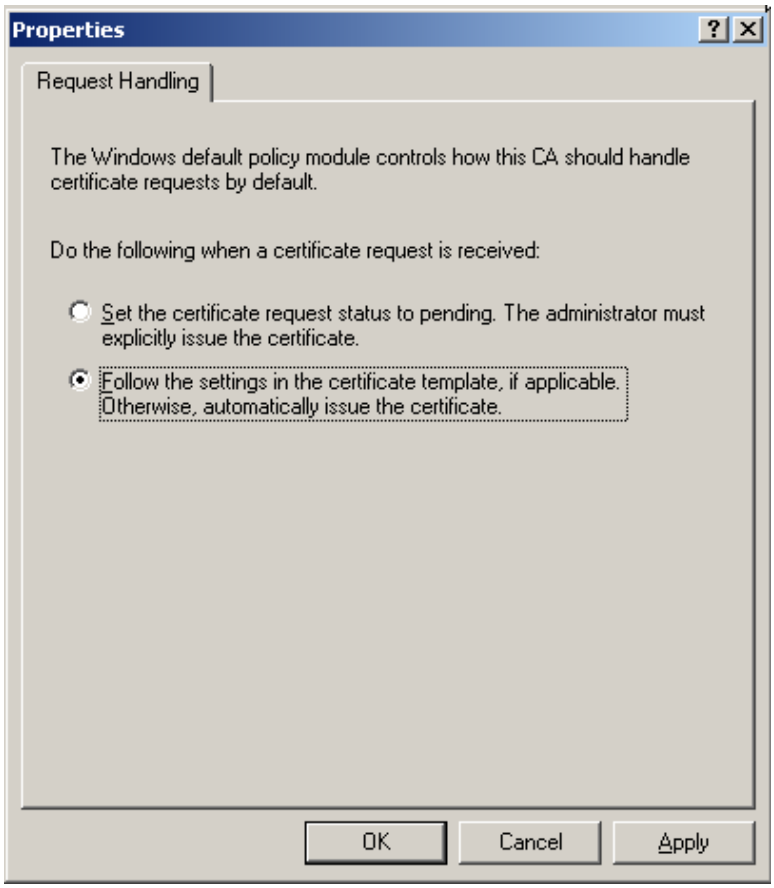

Рисунок 30

Шаг 20: в окне Windows default выдается предупреждение о необходимости перезапуска сертификатного сервиса:

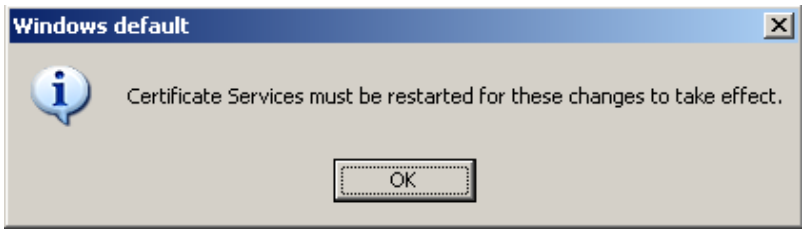

Рисунок 31

Шаг 21: в окне Certificate Authority выберите предложение меню Action, в выпадающем меню предложение All Tasks, а в следующем выпадающем меню – предложение `Stop Service` (Рисунок 32). После остановки сервиса выберите предложение `Start Service`.

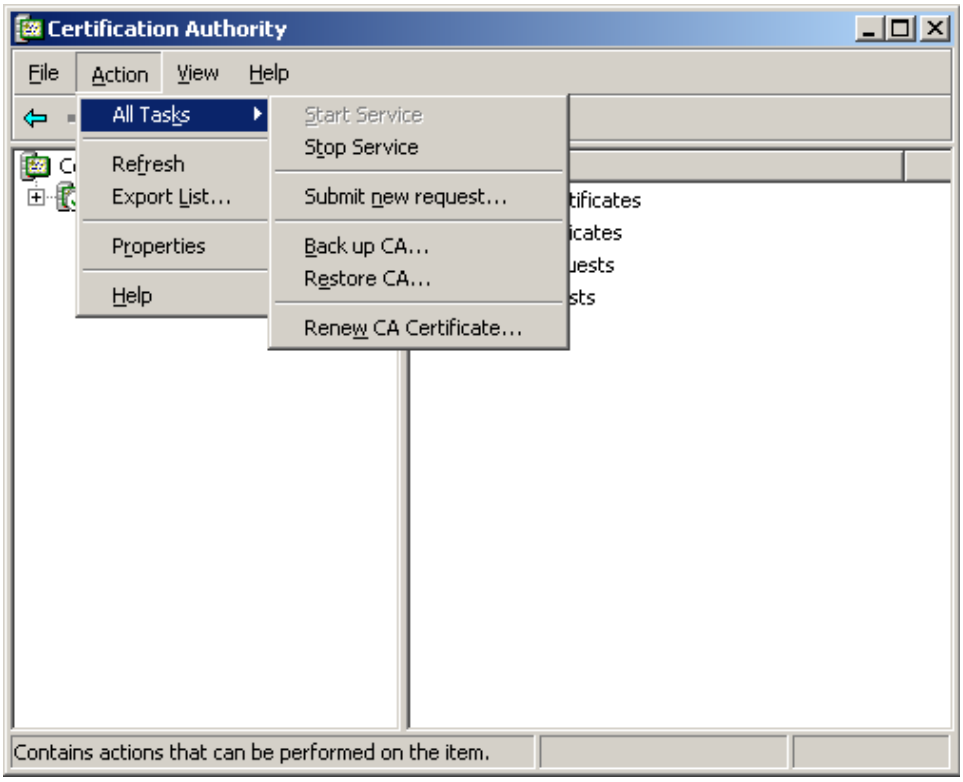

Рисунок 32

На этом создание Удостоверяющего Центра и его CA сертификата закончено.

Примечание:

Если была установлена версия 3.6 СКЗИ “КриптоПро CSP”, то для возможности дальнейшего выбора криптопровайдера “КриптоПро CSP” в окне создания запроса на сертификат, выполните следующее:

в файле `System32\certsrv\certsgcl.inc` измените значение константы `Const nMaxProvType` с 25 на 99. В стандартном скрипте перечисляются только 25 типов криптопровайдера, а “КриптоПро CSP” имеет тип 77.

Создание ключевой пары и формирование запроса на локальный сертификат

Опишем создание ключевой пары и формирование запроса на создание локального сертификата средствами Microsoft Windows с использованием алгоритмов ГОСТ на отдельном компьютере с установленной ОС Microsoft Windows, разместив контейнер с секретным ключом локального сертификата в Реестре.

Шаг 1: установите программный Продукт СКЗИ "КриптоПро CSP 3.6". Установка этого Продукта описана в разделе "Установка СКЗИ “КриптоПро CSP”.

Шаг 2: инсталлируйте ключевой носитель, на котором будет размещен контейнер с секретным ключом локального сертификата, например Реестр, используя СКЗИ "КриптоПро CSP 3.6". Эта инсталляция описана в разделе "Инсталляция ключевого считывателя Реестр".

Шаг 3: запустите `Microsoft Internet Explorer`. В поле `Address` укажите `IP`-адрес сервера Удостоверяющего Центра и запустите утилиту `certsrv (Certificate Service)`, например, `http://10.0.232.7/certsrv/`.

Создание Удостоверяющего Центра MS CA описано в разделе «Установка и настройка Удостоверяющего Центра. Создание CA сертификата». В качестве криптопровайдера на сервере устанавливается продукт СКЗИ "КриптоПро CSP 3.6".

Шаг 4: в появившемся окне высвечивается имя Удостоверяющего Центра – в нашем случае S-Terra Center CA. Для формирования запроса на создание локального сертификата выберите предложение `Request a certificate` (Рисунок 33):

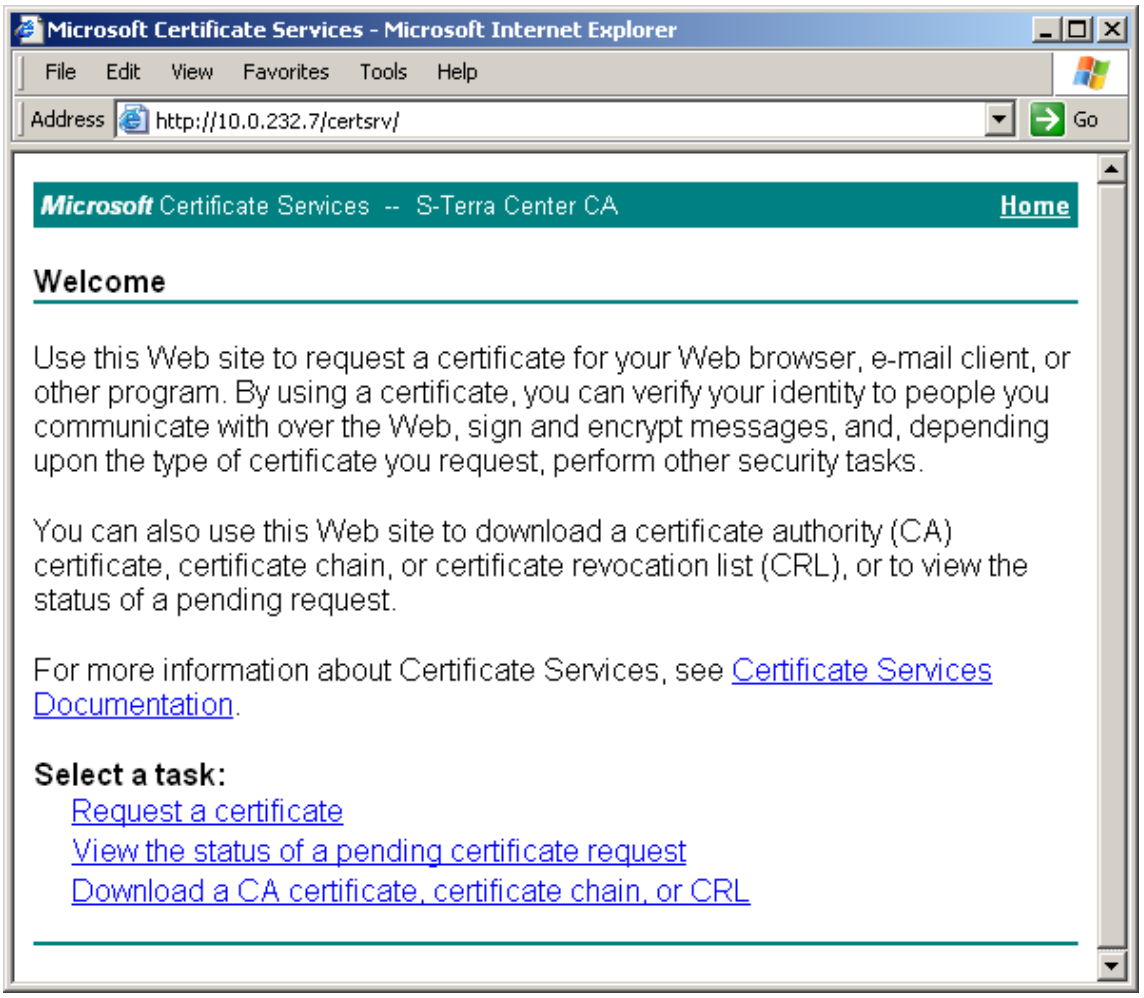

Рисунок 33

Шаг 5: выберите форму расширенного запроса – предложение “`advanced certificate request`”:

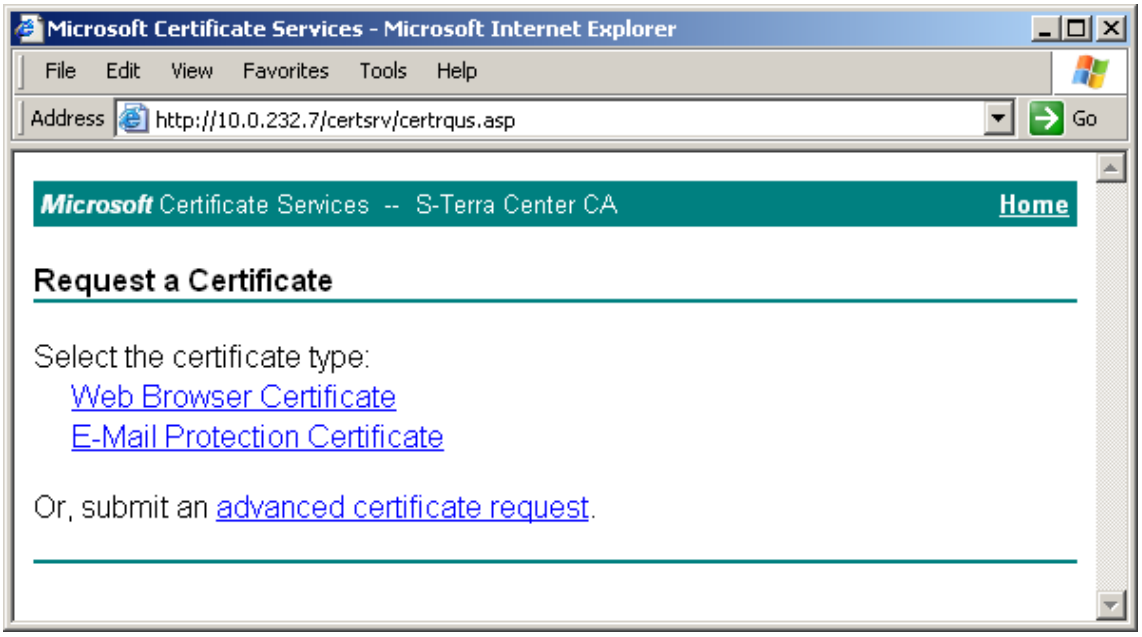

Рисунок 34

Шаг 6: для получения формы для формирования запроса на сертификат выберите предложение "`Create and submit a request to this CA`"t:

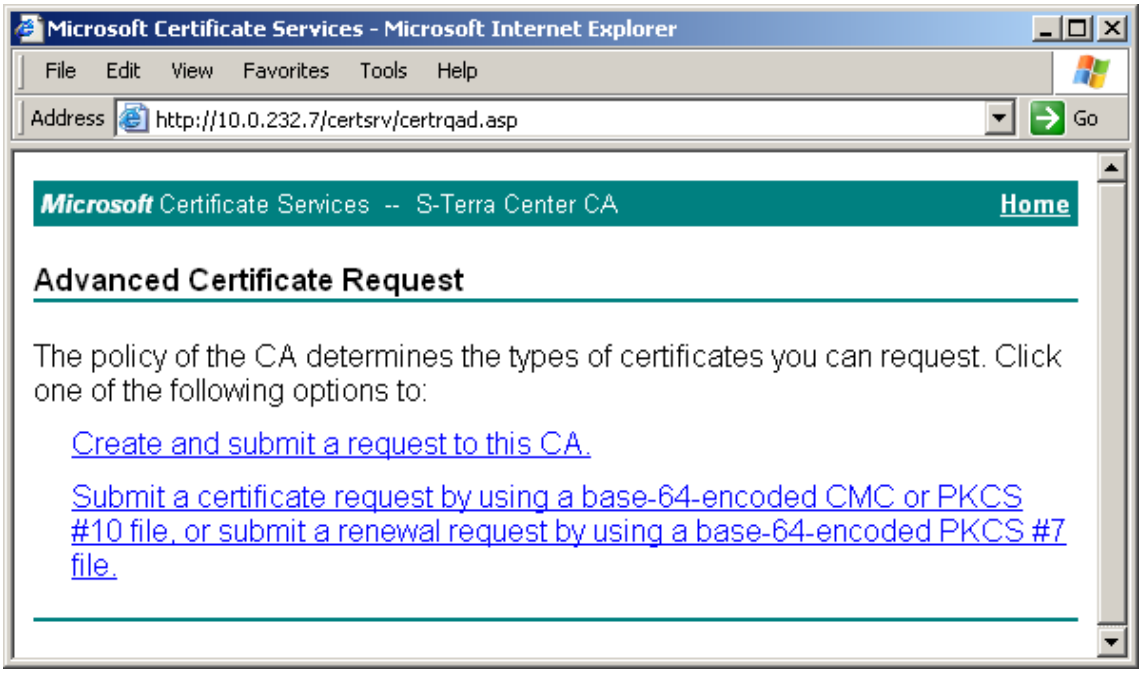

Рисунок 35

Шаг 7: заполните форму расширенного запроса, показанную ниже (Рисунок 36). Дадим некоторые пояснения для ее заполнения:

- в разделе **Identifying Information** (Информация о Владельце Сертификата) внесите данные о владельце сертификата. Во всех полях этого раздела разрешается использовать не только латинские, но и русские буквы, кроме поля Country/Region, оно всегда содержит значение RU.
  Примечание: если при создании запроса на сертификат при заполнении полей сертификата используются русские буквы, необходимо, чтобы они были введены в формате UTF-8
- в разделе Type of Certificate Needed (Тип требуемого сертификата) из выпадающего списка выберите предложение Client Authentication Certificate
- в разделе Key Options (Опции создания ключей) выбираются опции для создания ключевой пары и размещения секретного ключа. Рекомендуется сделать следующий выбор:
  - Поставьте переключатель в положение Create new key set (Создать установки для нового секретного ключа)
  - CSP (Тип Криптопровайдера) – из выпадающего списка выберите Crypto-Pro GOST R 34.10-2001 KC1 CSP
  - Key Usage (Использование ключей) – для выбора типа ключа поставьте переключатель в одно из трех положений: Signature (для подписи), Exchange (для обмена), Both (для подписи и обмена)
  - Key Size (Размер ключа) – размер ключа. При выборе алгоритма GOST R 34.10-2001 длина ключа всегда 512
  - поставьте переключатель в положение User specified key container name, чтобы задать имя контейнера с секретным ключом
  - в поле Container name (Имя контейнера) введите имя контейнера, в котором будет размещен секретный ключ без указания ключевого носителя, выбрать ключевой носитель будет предложено далее. В имени контейнера разрешается использовать латинские буквы и цифры
  - Mark keys as exportable – поставьте флажок, чтобы можно было скопировать контейнер с секретным ключом с одного ключевого носителя на другой, а также во время создания инсталляционного файла провести проверку соответствия локального сертификата и секретного ключа
    - Export keys to file – этот флажок выставляется, если нужно экспортировать ключи в файл. Мы этот флажок не выставляем, так как секретный ключ размещаем в контейнере
  - Store certificate in the local computer certificate store (Использовать локальное хранилище) – всегда выставляйте этот флажок
- в разделе Additional Options (Дополнительные опции):
  - Hash Algorithm – выбрать GOST R34.11-94
  - далее установок никаких делать не нужно.

По этому образцу заполните форму запроса и нажмите кнопку Submit (послать запрос):

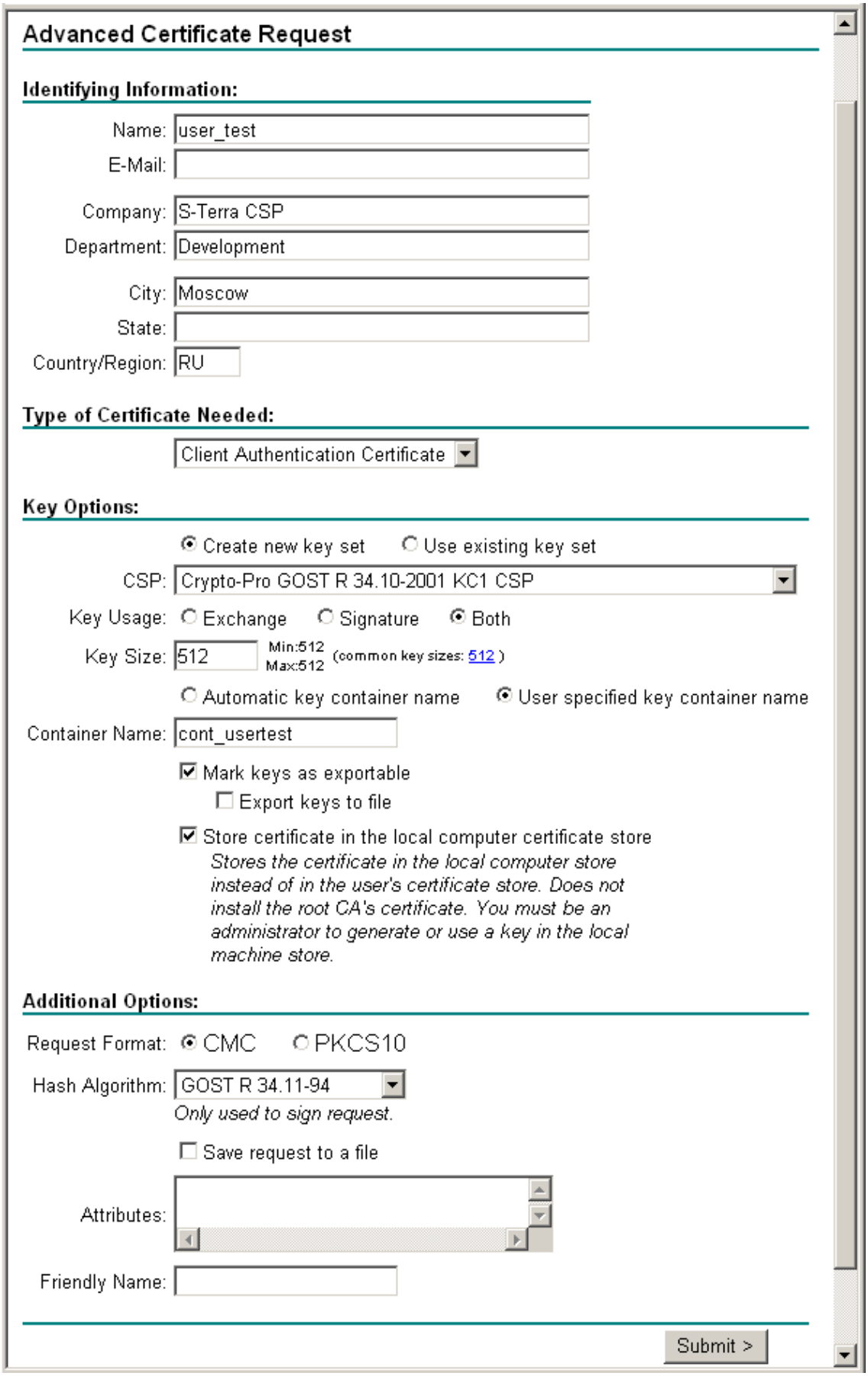
Advanced Certificate Request
Identifying Information:
Name: user_test
E-Mail:
Company: S-Terra CSP
Department: Development
City: Moscow
State:
Country/Region: RU
Type of Certificate Needed:
Client Authentication Certificate
Key Options:
Create new key set
Use existing key set
CSP: Crypto-Pro GOST R 34.10-2001 KC1 CSP
Key Usage: Exchange Signature Both
Key Size: 512 Min:512 Max:512 (common key sizes: 512)
Automatic key container name
User specified key container name
Container Name: cont_usertest
Mark keys as exportable
Export keys to file
Store certificate in the local computer certificate store
Stores the certificate in the local computer store instead of in the user's certificate store. Does not install the root CA's certificate. You must be an administrator to generate or use a key in the local machine store.
Additional Options:
Request Format: CMC PKCS10
Hash Algorithm: GOST R 34.11-94
Only used to sign request.
Save request to a file
Attributes:
Friendly Name:
Submit >

Рисунок 36

Шаг 8: появляется предупреждение (Рисунок 37), нажмите кнопку Yes, чтобы продолжить:

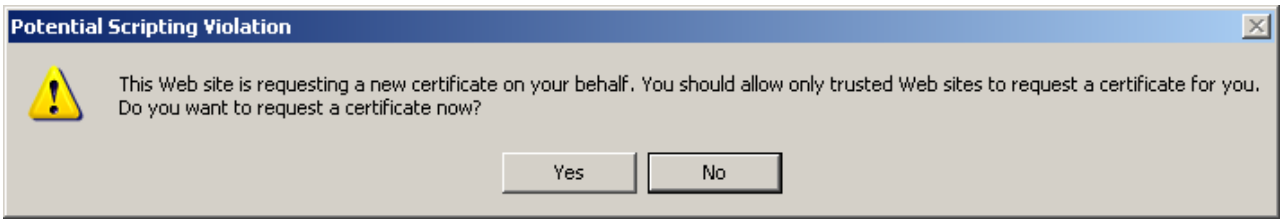

Рисунок 37

Шаг 9: выберите ключевой носитель, например Реестр, для размещения контейнера с секретным ключом и нажмите ОК.

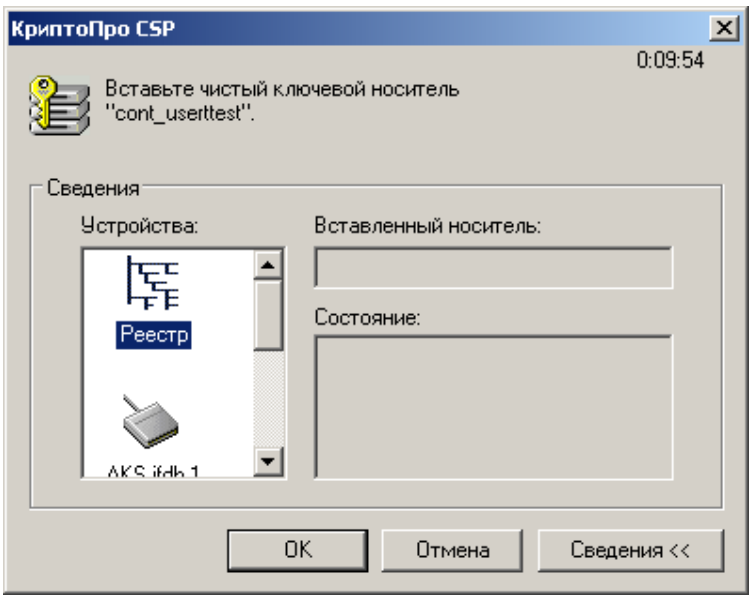

Рисунок 38

Шаг 10: для создания ключевой пары генератор случайных чисел просит нажать любую клавишу или подвигать мышкой:

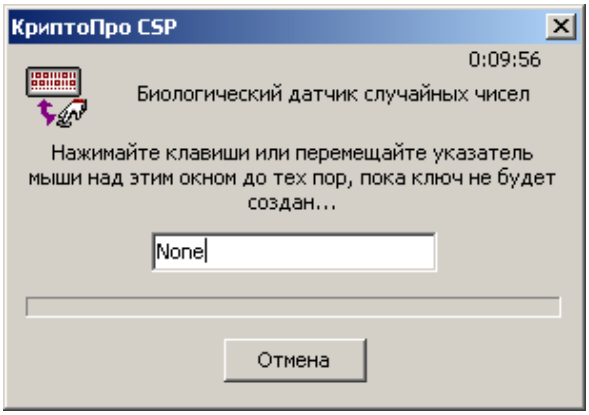

Рисунок 39

Шаг 11: задайте пароль на контейнер с секретным ключом и нажмите ОК:

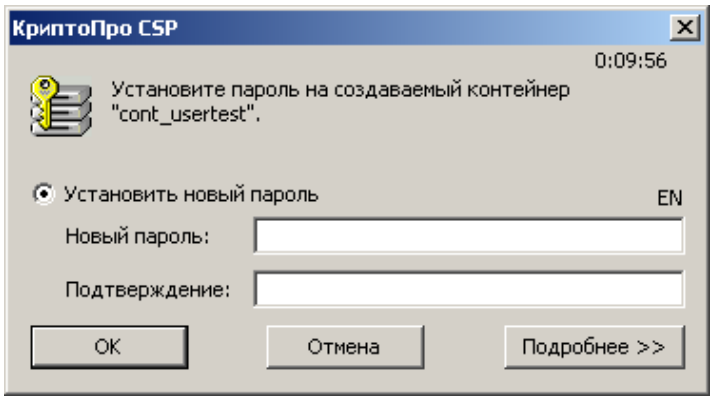

Рисунок 40

Таким образом, ключевая пара – открытый и секретный ключи созданы. Секретный ключ размещен в контейнере на ключевом носителе Реестр и защищен паролем. А на основе открытого ключа Удостоверяющий Центр создаст локальный сертификат.

Шаг 12: Удостоверяющий Центр сразу создал локальный сертификат и прислал об этом уведомление. При выборе предложения `Install this certificate` , сертификат будет получен из Удостоверяющего Центра и размещен в контейнере с секретным ключом, в нашем примере – в Реестре.

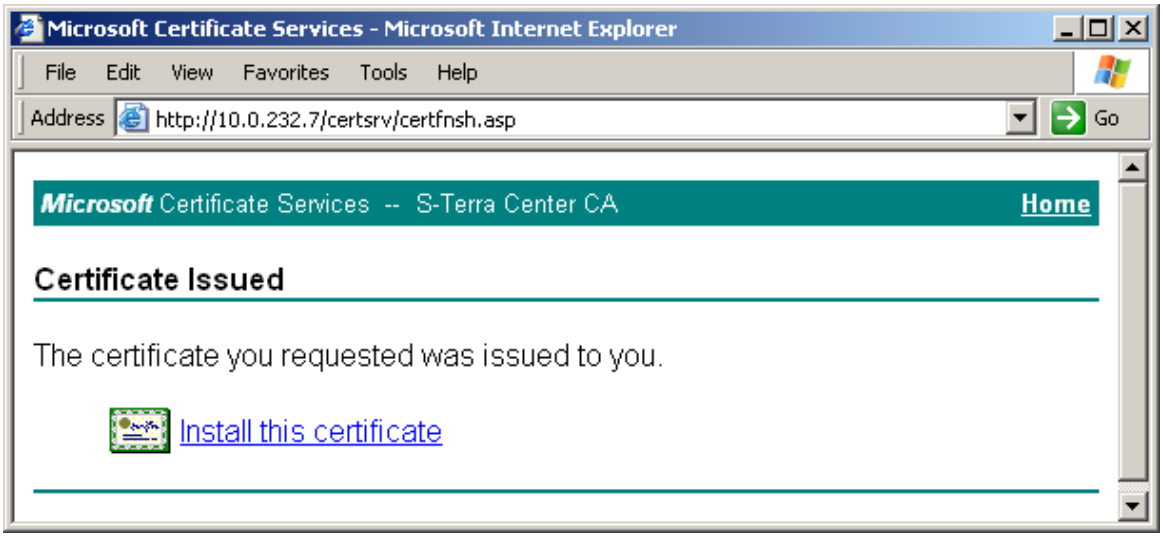

Рисунок 41

Шаг 13: появляется предупреждение (Рисунок 42), нажмите кнопку Yes, чтобы продолжить:

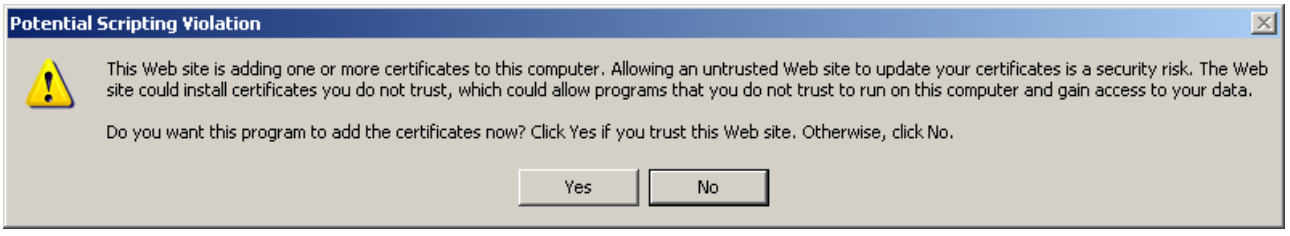

Рисунок 42

Шаг 14: еще раз введите пароль на контейнер с секретным ключом и нажмите ОК

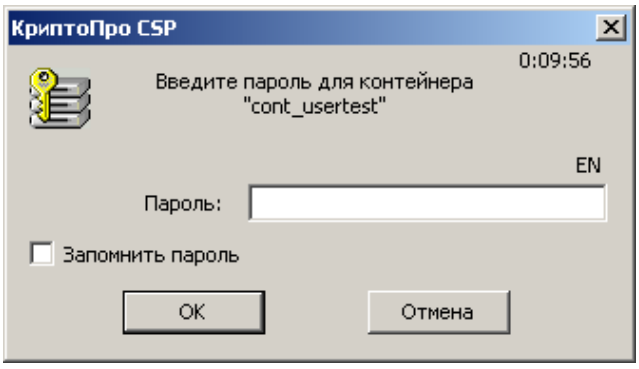

Рисунок 43

После размещения локального сертификата в контейнер выдается сообщение:

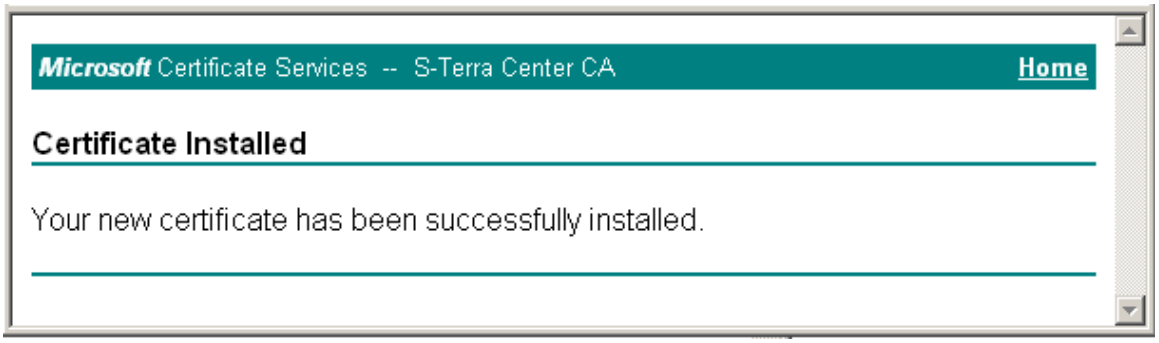

Рисунок 44

Локальный сертификат можно получить из Удостоверяющего Центра и другим способом, но приведенный выше наиболее удобен.

Для регистрации локального сертификата на шлюзе безопасности необходимо экспортировать локальный сертификат из контейнера в файл, поэтому перейдите к следующему разделу.

Экспортирование локального сертификата в файл

Для экспортирования локального сертификата из контейнера в файл выполните следующие действия:

Шаг 1: запустите продукт "КриптоПро CSP 3.6" – Пуск – Настройка – Панель управления – КриптоПро CSP.

Шаг 2: войдите во вкладку Сервис и нажмите кнопку Просмотреть сертификаты в контейнере…

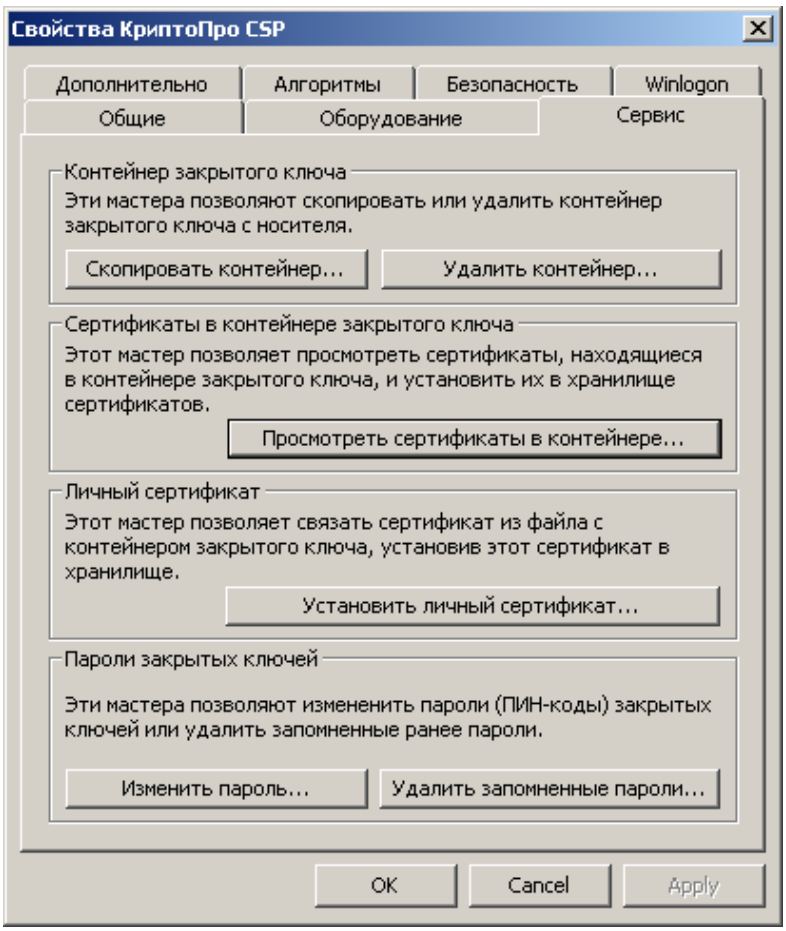

Рисунок 45

Шаг 3: в следующем окне для указания контейнера поставьте переключатель в положение Компьютера и нажмите кнопку Обзор…

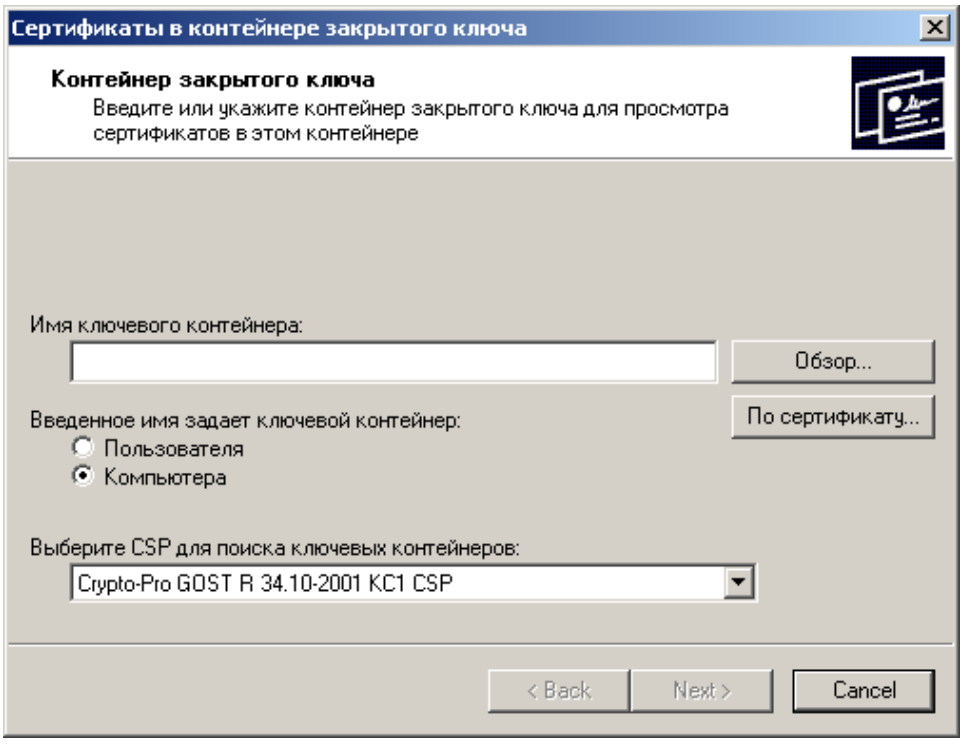

Рисунок 46

Шаг 4: в окне со списком контейнеров, размещенных в Реестре, поставьте переключатель в положение Уникальные имена и выберите контейнер, в котором лежит секретный ключ и сертификат пользователя, и нажмите кнопку ОК:

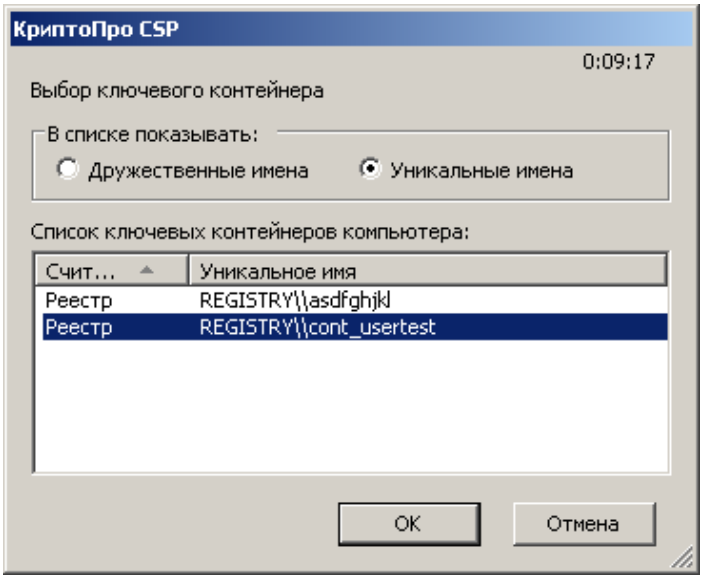

Рисунок 47

Шаг 5: выбор контейнера произведен, нажмите кнопку Next:

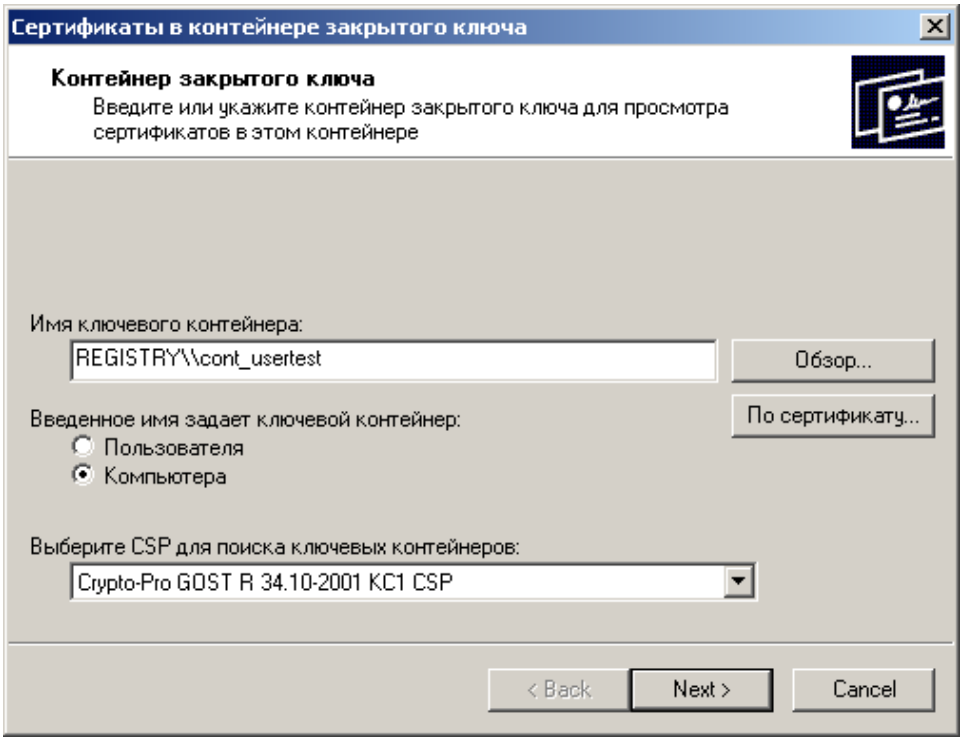

Рисунок 48

Шаг 6: следующее окно показывает поля локального сертификата, нажмите кнопку Свойства:

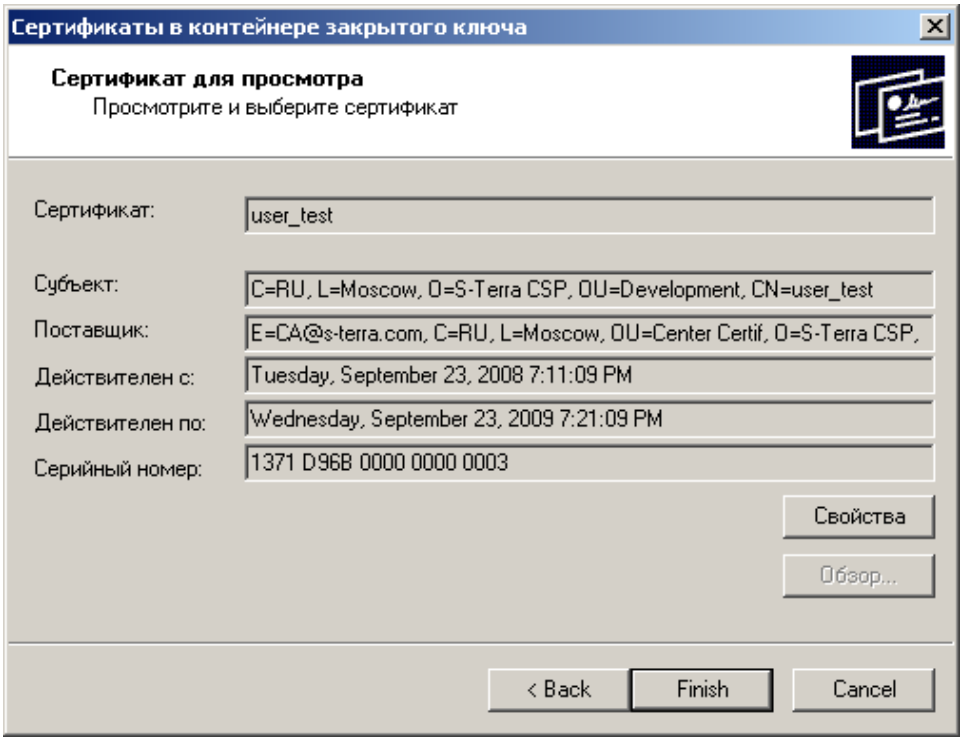

Рисунок 49

Шаг 7: в окне Property Page Select Cert выберите вкладку Detail и нажмите кнопку Copy to File…

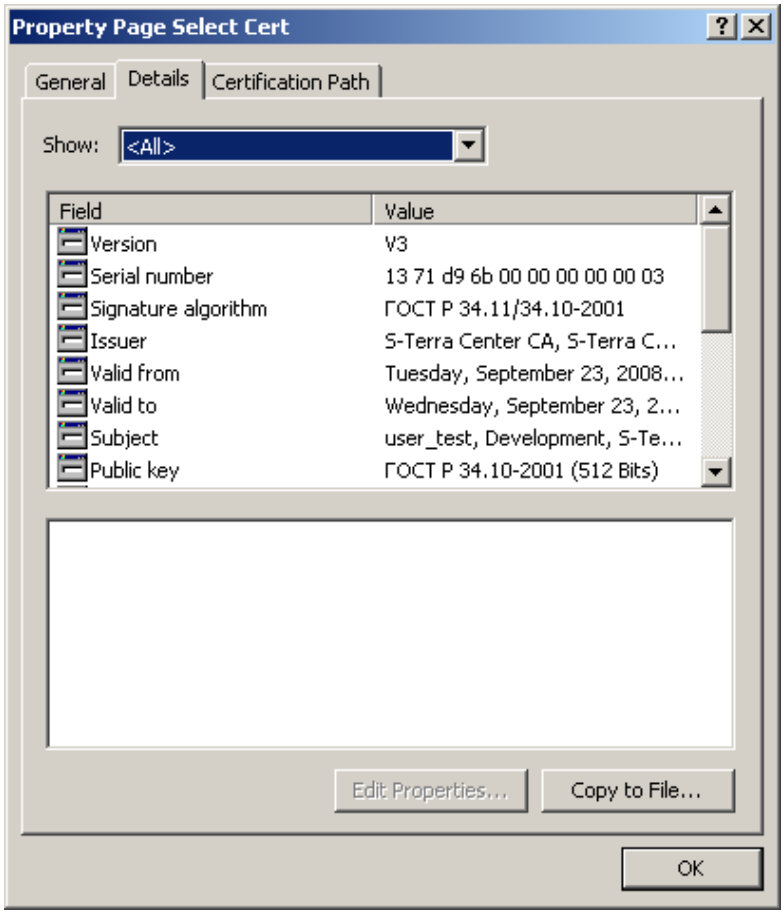

Рисунок 50

Шаг 8: в окне визарда нажмите кнопку Next:

Рисунок 51

Шаг 9: установите переключатель во второе положение, чтобы экспортировать в файл только сертификат без секретного ключа и нажмите Next:

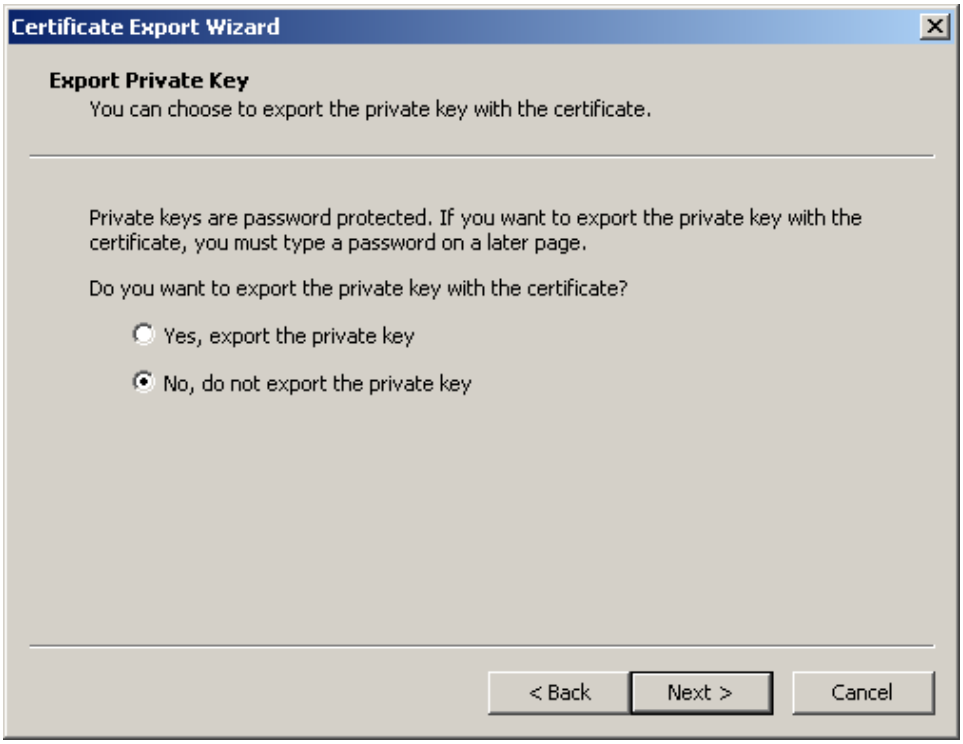

Рисунок 52

Шаг 10: выберите формат файла сертификата – DER encoded binary X.509 (.CER) и нажмите Next:

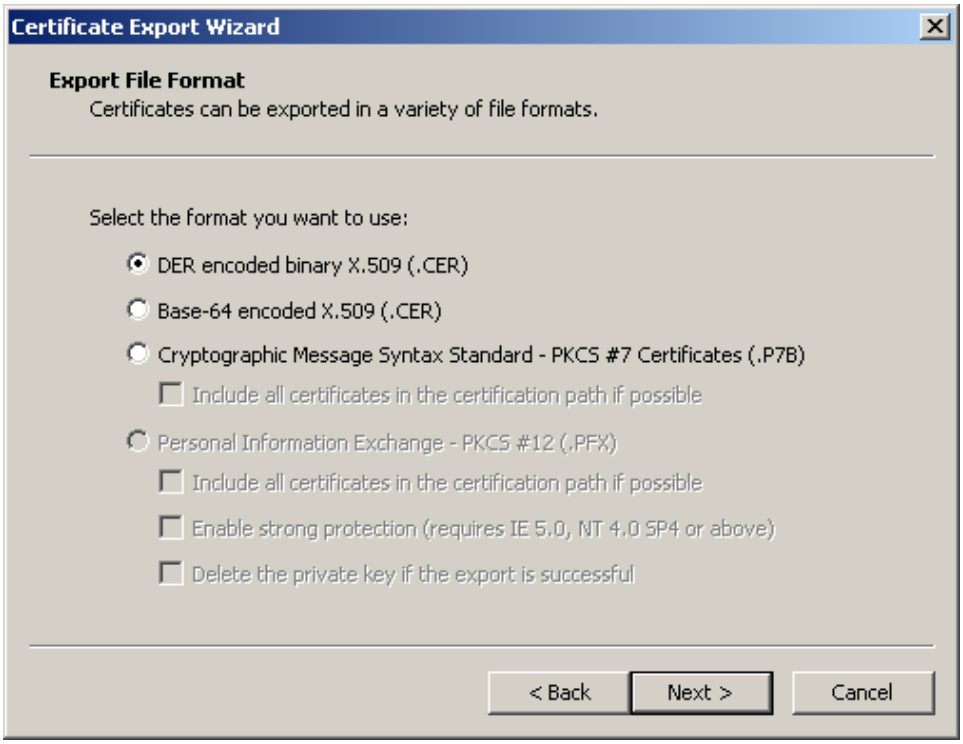

Рисунок 53

Шаг 11:укажите имя файла, в который экспортируется сертификат, и нажмите Next:

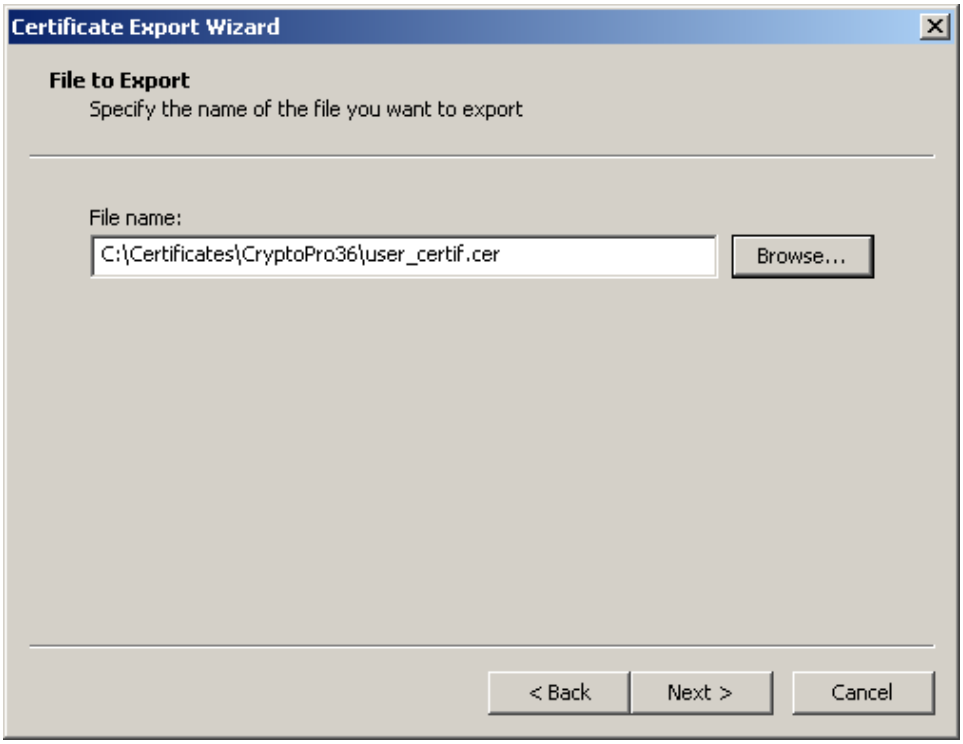

Рисунок 54

Шаг 12: экспортирование локального сертификата в файл закончено, нажмите Finish.

Рисунок 55

На этом создание локального сертификата для шлюза безопасности и CA сертификата закончено, они оба экспортированы в файл.

Доставьте оба сертификата на шлюз безопасности любым доступным способом и зарегистрируйте их. Контейнер с секретным ключом локального сертификата можно скопировать на дискету и доставить на шлюз безопасности.

Создание ключевой пары и запроса на локальный сертификат с помощью утилиты cryptcp

В состав поставляемого продукта CSP VPN Gate входит пакет «КриптоПро CSP 3.6» с утилитой `cryptcp`, созданный компанией "Крипто-Про".

Утилита `cryptcp` размещена в каталоге:

`/opt/cprocsp/bin/ia32` (в ОС Red Hat Enterprise Linux 5)

`/opt/cprocsp/bin/ia32` (в ОС Solaris 10).

Для создания ключевой пары и формирования запроса на локальный сертификат выполните следующие команды:

`cd /opt/cprocsp/bin/ia32` (в ОС Red Hat Enterprise Linux 5)

или

`cd /opt/cprocsp/bin/ia32` (в ОС Solaris 10)

```
./cryptcp -creatrqst -dn 'C=RU,O=S-Terra,OU=QA,CN=g100' -provtype 75 -
both -km -cont '\\.\HDIMAGE\g100' /tmp/g100.req
```

где

`dn` поля сертификата

`provtype 75` тип криптопровайдера, по умолчанию 75 – Crypto-Pro GOST R34.10-2001 CSP (71 - Crypto-Pro GOST R34.10-94 CSP)

`both` создается два типа ключей - для подписи и шифрования

`km` разместить контейнер на компьютере

`HDIMAGE` имя считывателя (жесткий диск)

`g100` имя контейнера

`/tmp/g100.req` имя файла с запросом на сертификат в формате PKCS#10.

Созданный таким образом контейнер с секретным ключом никуда экспортировать не нужно – он находится на шлюзе безопасности, а запрос на локальный сертификат нужно отослать в Удостоверяющий Центр, имеющий CA сертификат, созданный с использованием криптопровайдера “КриптоПро CSP 3.6”. Процедура отсылки запроса через Web-интерфейс Удостоверяющего Центра описана в разделе “Создание локального сертификата” и совпадает с отсылкой запроса, созданного с использованием криптопровайдера “Signal-COM CSP”.

Создание локального сертификата с использованием "Signal-COM CSP"

Для создания локального сертификата для шлюза безопасности с использованием криптопровайдера "Signal-COM CSP", который можно использовать для работы с продуктами CSP VPN Agent, опишем план действий:

- установить криптопровайдера "Signal-COM CSP"
- инсталлировать и настроить Удостоверяющий Центр (Certification Authority). Описано как установить и настроить Microsoft Certification Authority, и создать CA сертификат
- установить Admin-PKI, реализующий российские криптографические алгоритмы. Создать ключевую пару и запрос на локальный сертификат.
- создать локальный сертификат.

Приведем подробные инструкции по каждому пункту.

Установка "Signal-COM CSP"

Для выполнения инсталляции необходимо:

- операционная система Windows 2000 Server (или выше)
- программный продукт "Signal-COM CSP" версии 1.407 (или выше)
- описание криптопровайдера "Signal-COM CSP" можно посмотреть по адресу: http://www.signal-com.ru/ru/prod/crypt/signal_com/
- демо-версию этого продукта можно запросить по адресу: http://www.signal-com.ru/ru/demo/csp/index.php

Запустите инсталляцию из файла `sccsp.exe` и следуйте инструкциям инсталлятора:

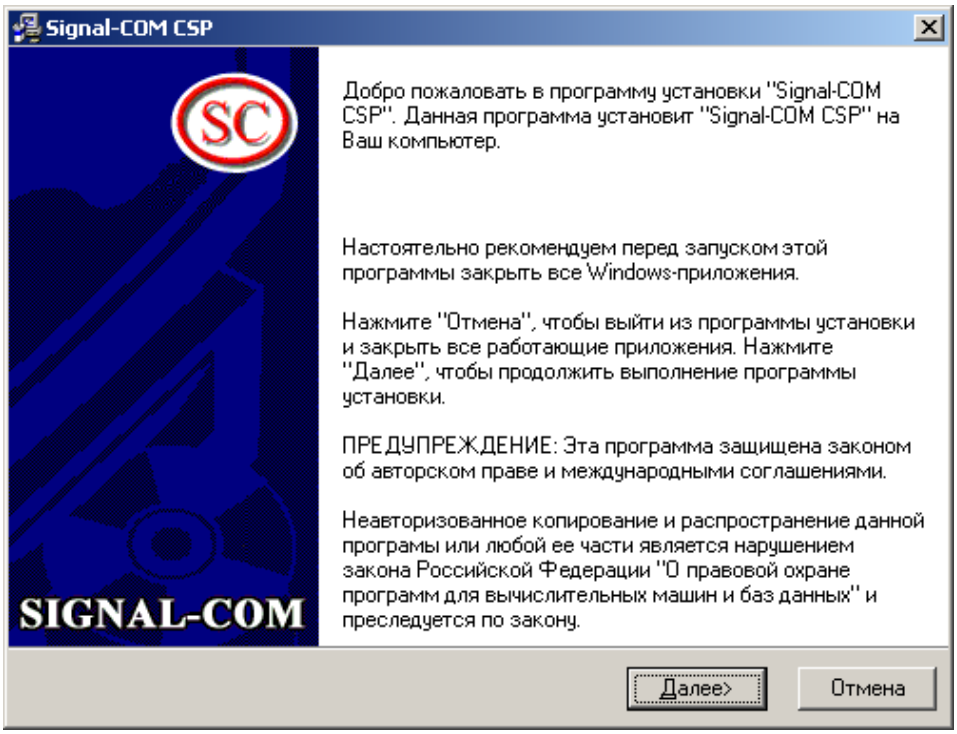

Рисунок 56

Введите требуемую информацию:

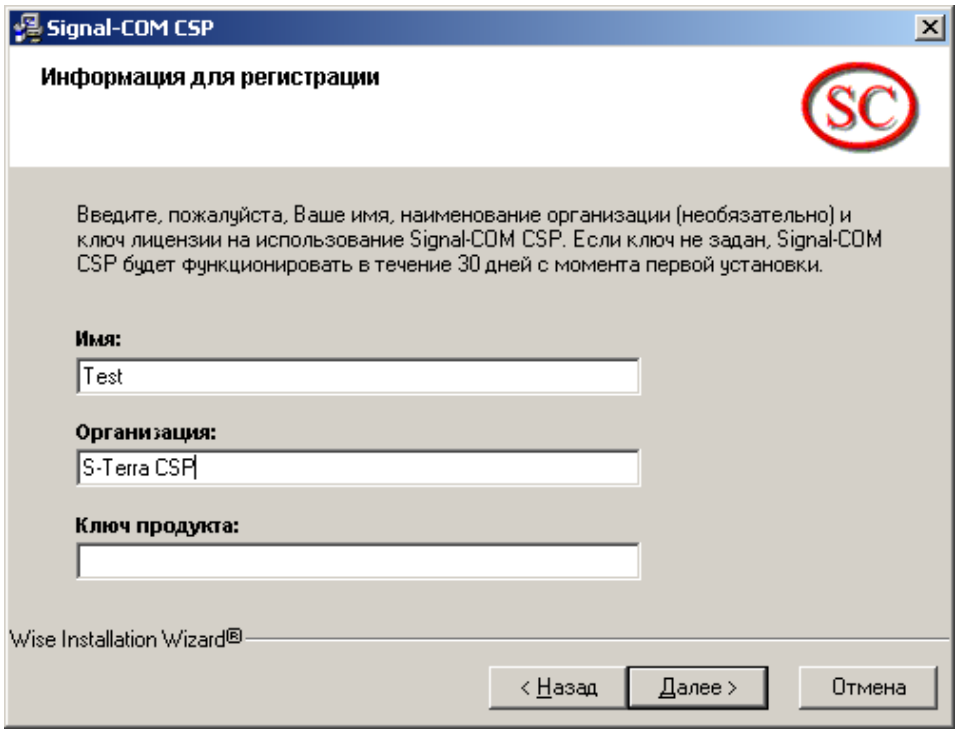

Рисунок 57

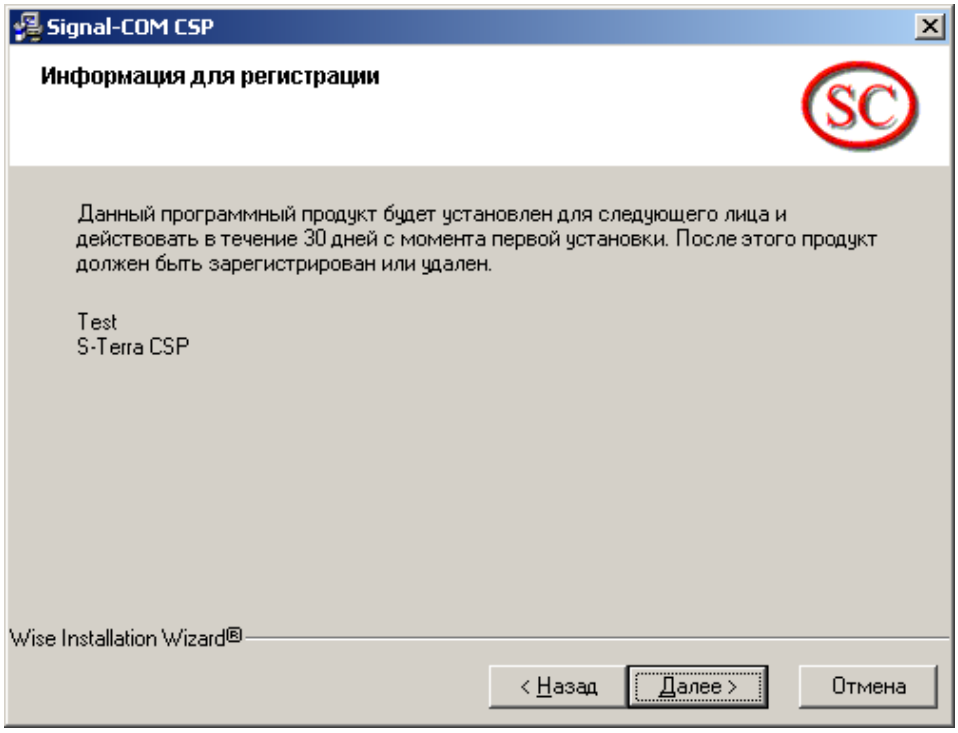

Рисунок 58

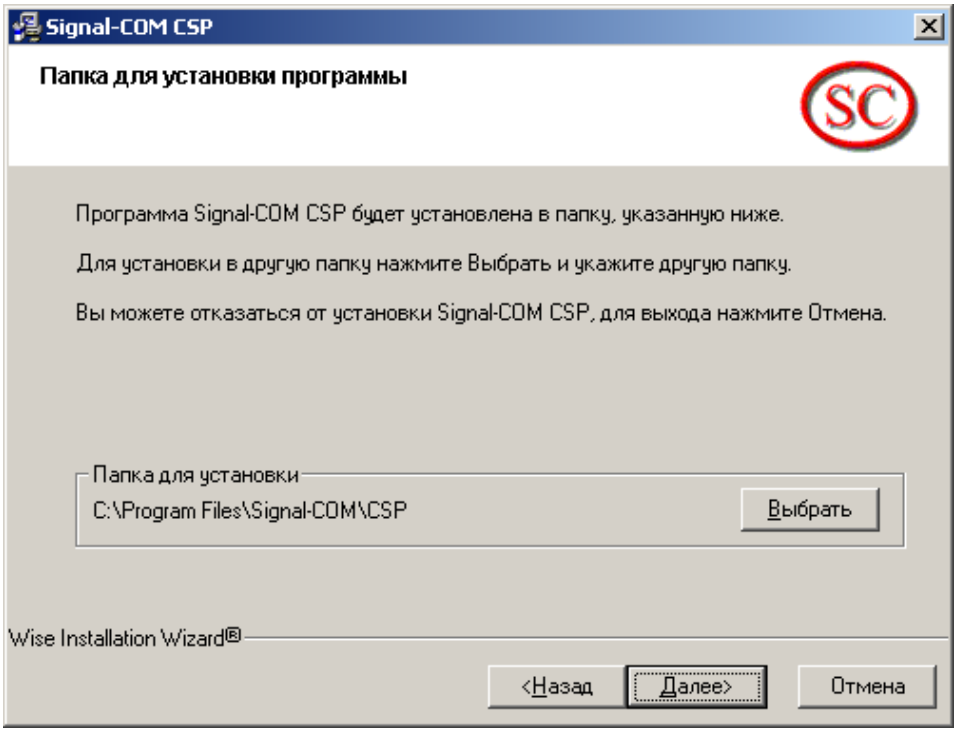

Рисунок 59

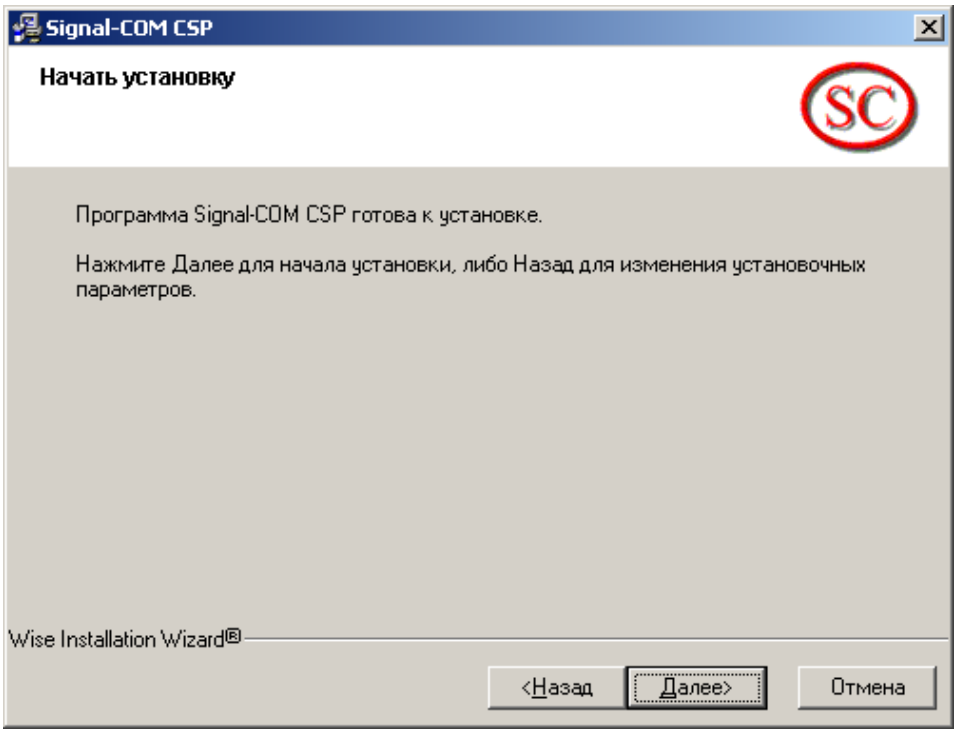

Рисунок 60

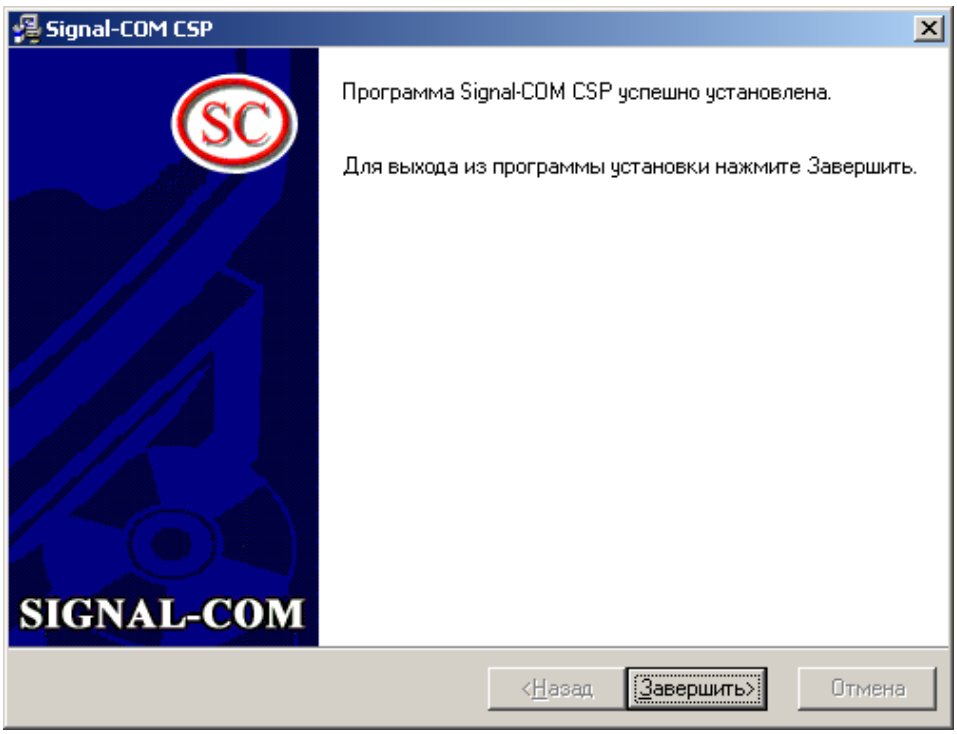

Рисунок 61

Инсталляция криптопровайдера завершена.

Установка и настройка Удостоверяющего Центра. Создание CA сертификата

Если вам известен Удостоверяющий Центр, в который вы будете отправлять запрос на локальный сертификат, то этот раздел можно пропустить и перейти к следующему, в противном случае – создайте Удостоверяющий Центр.

На компьютере с установленной ОС Windows 2003 Server для инсталляции Удостоверяющего Центра Microsoft Certification Authority выполните следующее:

Шаг 1: в окне установки компонент Windows (`Start-Settings-Control Panel-Add/Remove Programs-Add/Remove Windows Components`) установите флажок Application Server, а нажав на кнопку Details, установите сервисы Internet Information Services (IIS) и ASP.NET. Затем установите флажок `Certificate Services` и нажмите `Next` (Рисунок 62). Если `Certificate Services` уже установлен, то его нужно удалить (снять флажок), а потом снова установить:

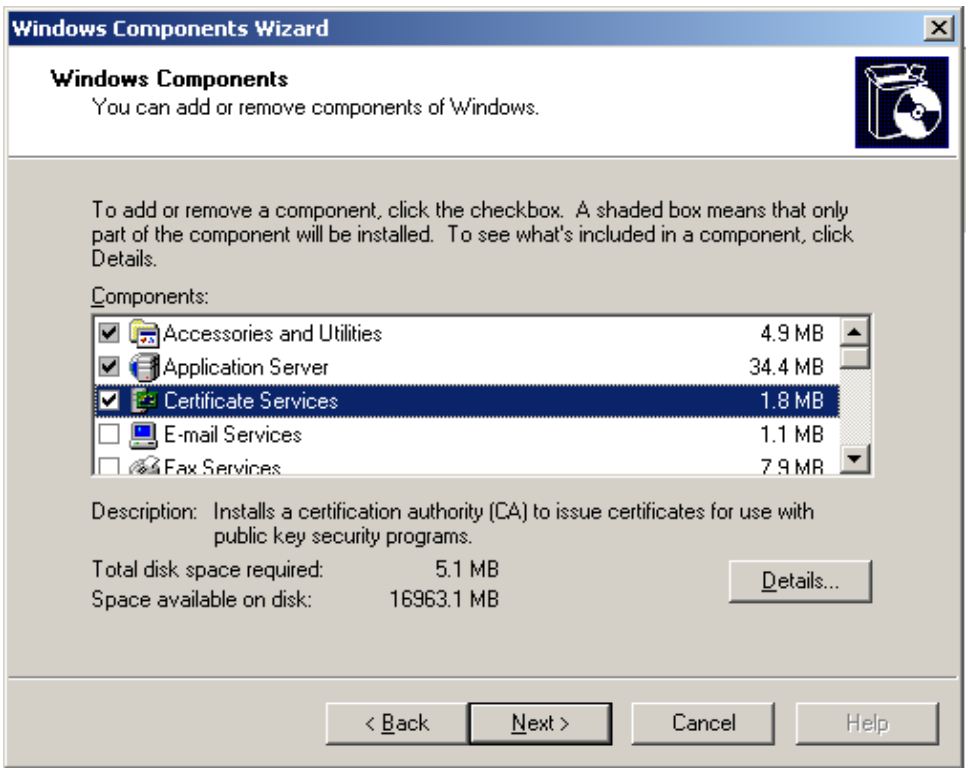

Рисунок 62

Перед установкой `Certificate Services` будет выдано предупреждение (Рисунок 63). Нажмите кнопку `Yes`.

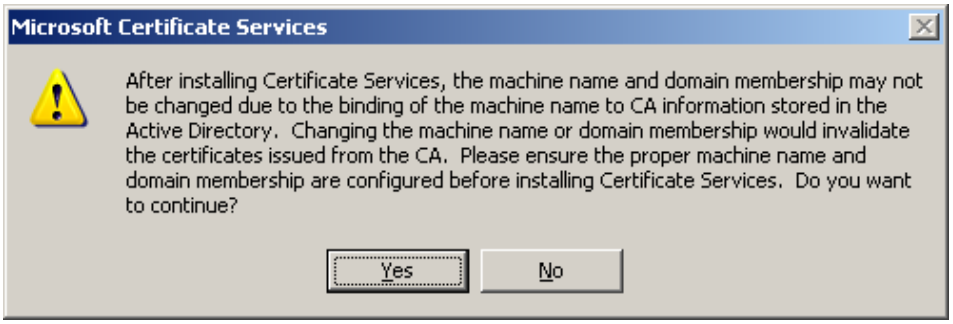

Рисунок 63

Шаг 2: выберите УЦ с корневым CA сертификатом: поставьте переключатель в положение `Stand-alone root CA`. Установите флажок `Use custom settings to generate the key pair and CA certificate` и нажмите `Next`:

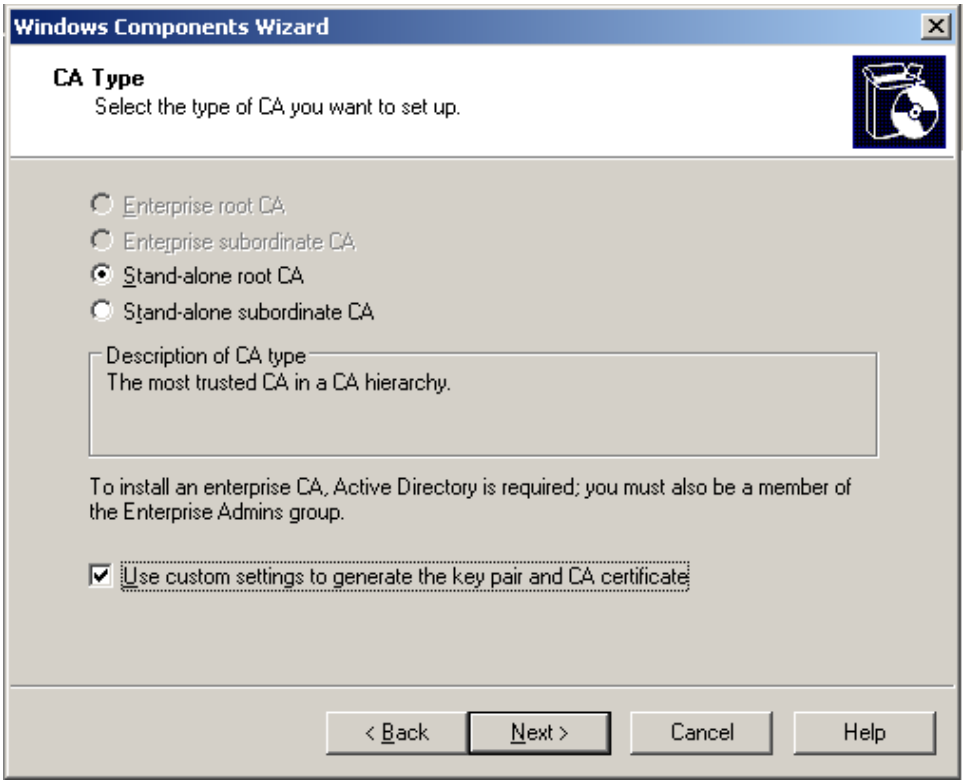

Рисунок 64

Шаг 3: в качестве криптопровайдера выберите `Signal-COM Enhanced Cryptographic Provider` (или `Signal-COM Special Cryptographic Provider` для использования CryptoPro совместимых алгоритмов):

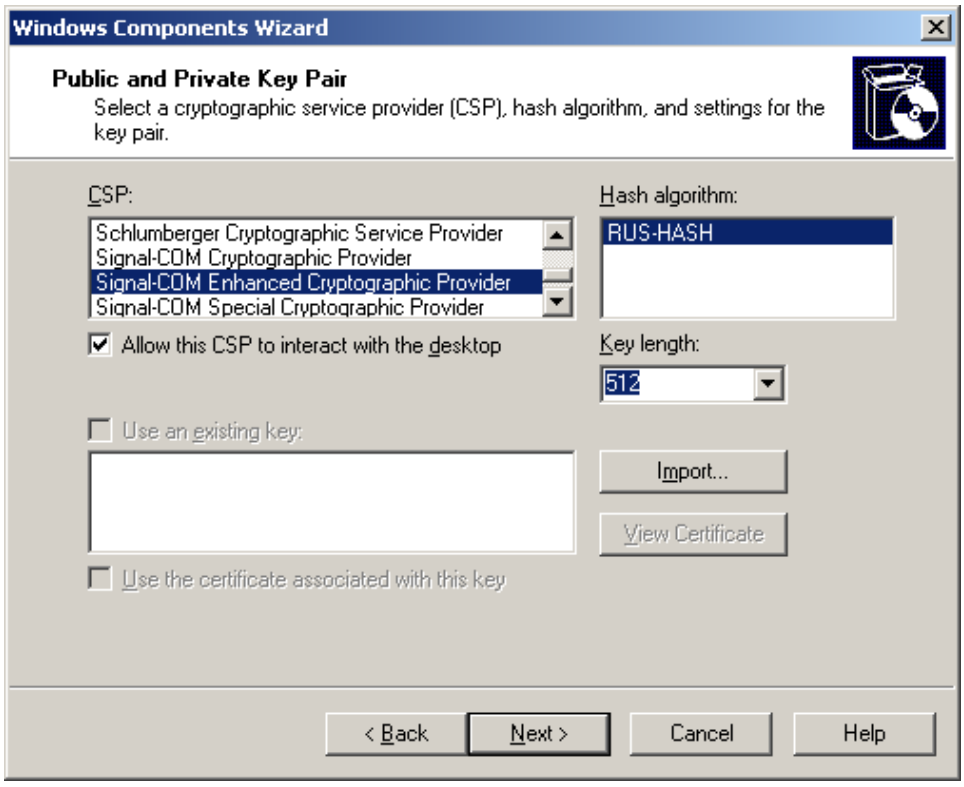

Рисунок 65

Шаг 4: заполните поля для CA сертификата и нажмите `Next`:

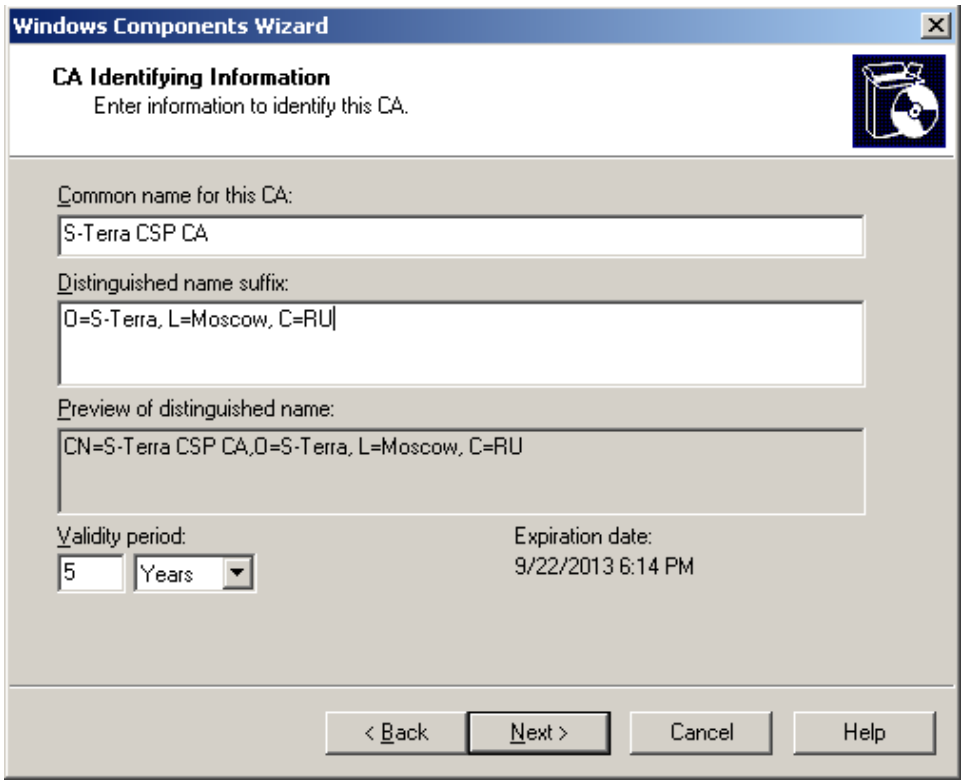

Рисунок 66

Шаг 5: выберите ключевой носитель - Реестр, на котором будет записан контейнер с секретным ключом для CA сертификата и нажмите `OK`:

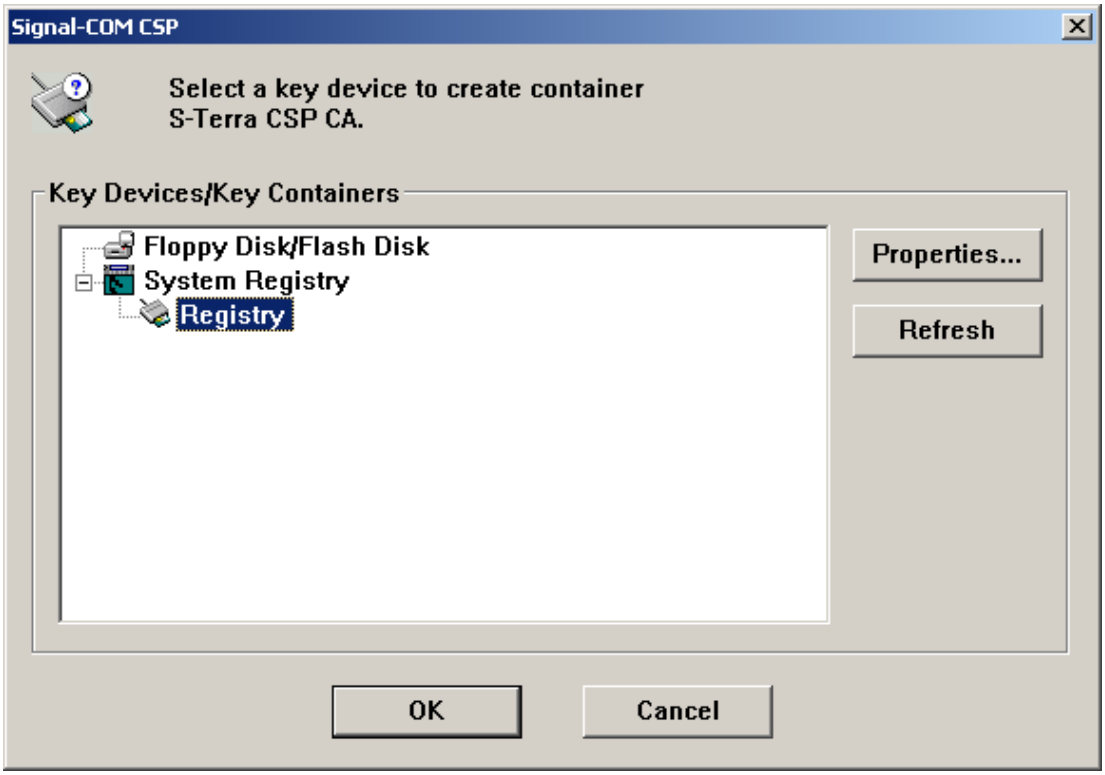

Рисунок 67

Шаг 6: далее происходит создание ключей для CA сертификата. На вопрос об использовании пароля к ключевому контейнеру нажмите `Yes`. Пароль рекомендуется, но не обязателен.

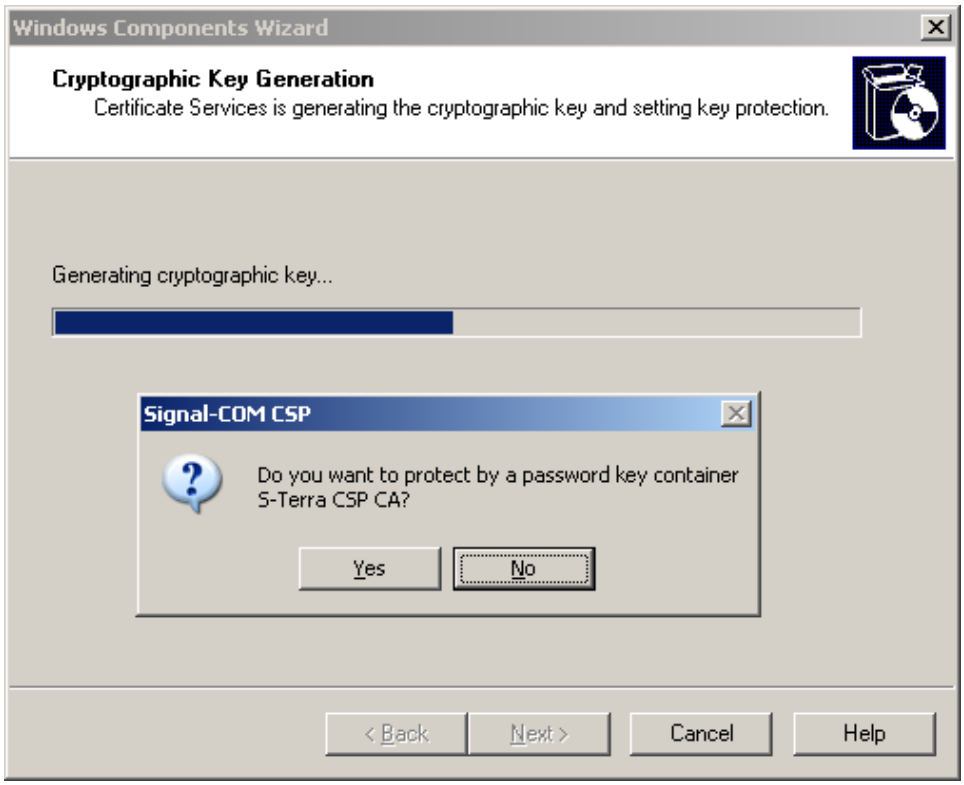

Рисунок 68

Шаг 7: задайте пароль к ключевому контейнеру и нажмите `OK`:

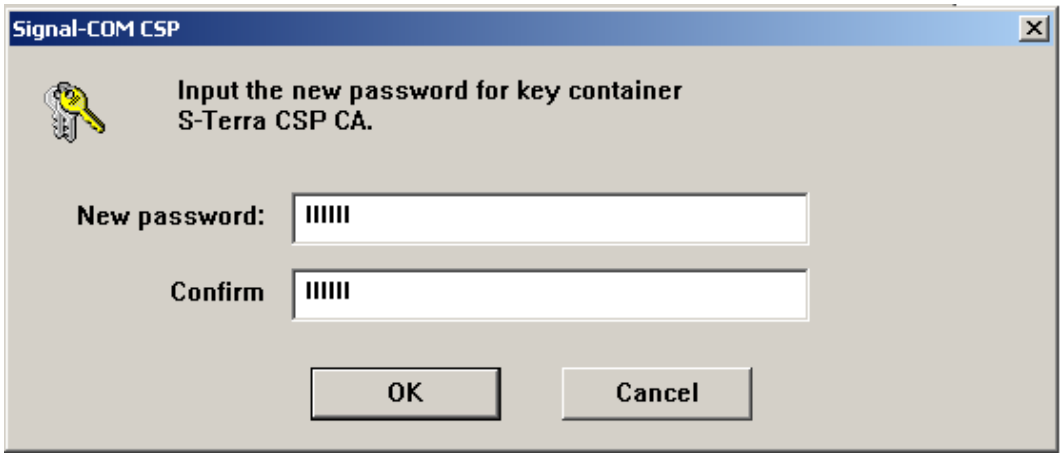

Рисунок 69

Шаг 8: для генератора случайных чисел вводите запрашиваемые символы:

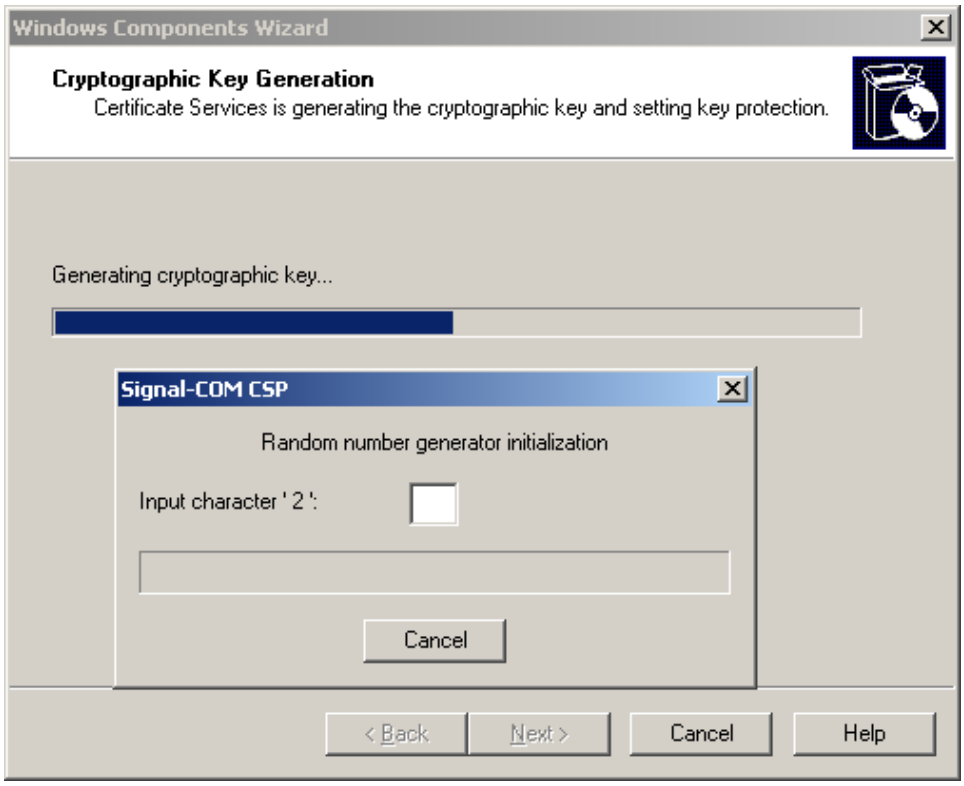

Рисунок 70

Шаг 9: оставьте без изменения установленные пути и нажмите Next:

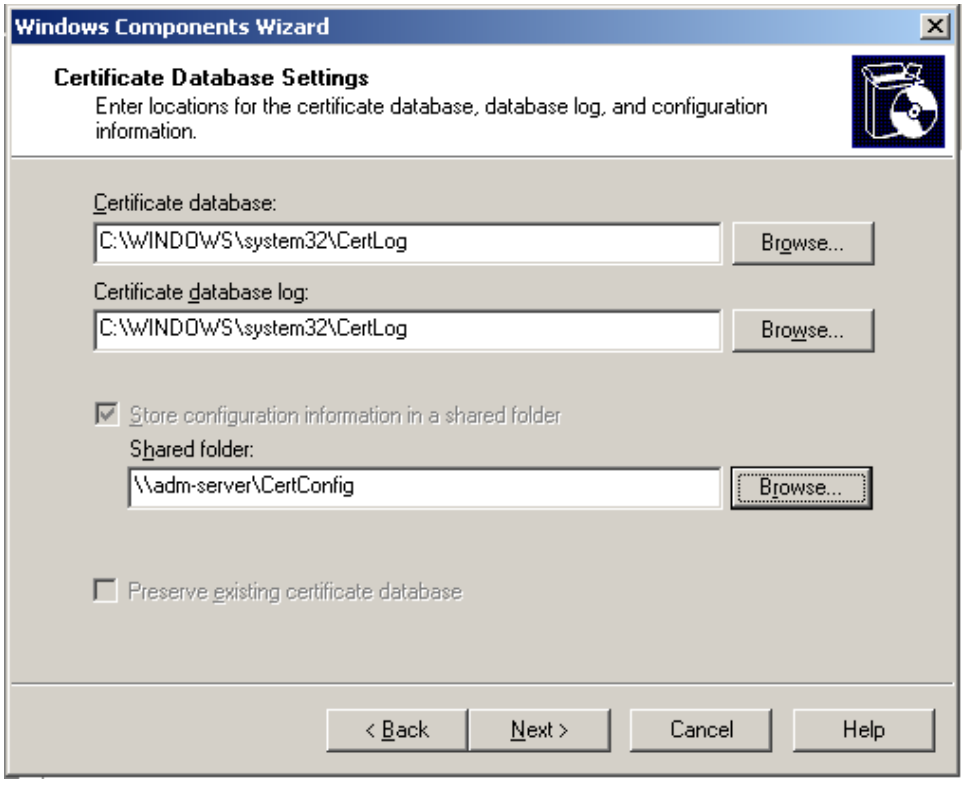

Рисунок 71

Шаг 10: введите еще раз пароль к контейнеру с секретным ключом CA сертификата и нажмите `OK`:

Рисунок 72

Шаг 11: инсталляция Удостоверяющего Центра завершена, нажмите Finish.

Рисунок 73

Шаг 12: для экспортирования CA сертификата в файл войдите в Certificate Authority (Start-Settings-Control Panel-Administrative Tools-Certificate Authority), выберите центр сертификации, а затем Action - Properties:

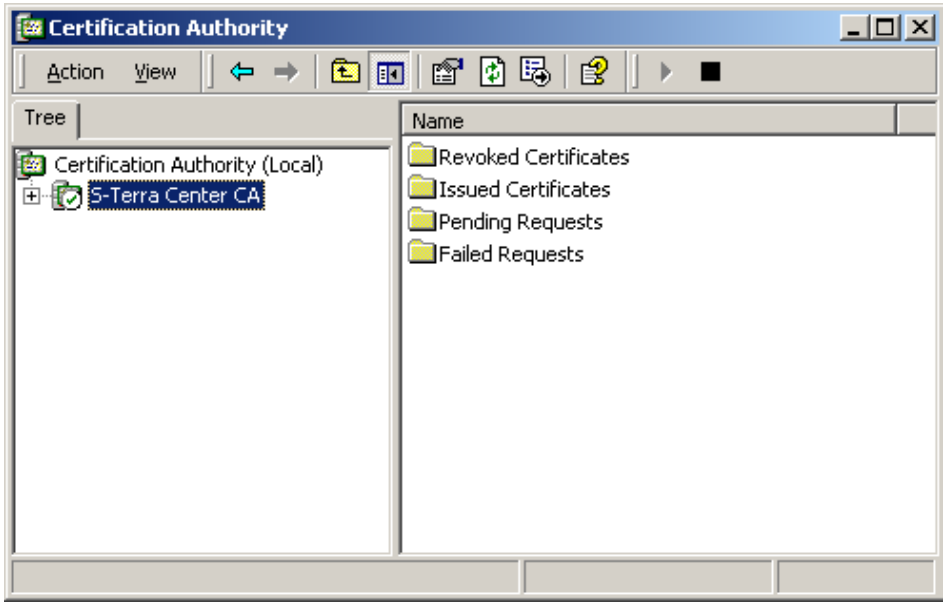

Рисунок 74

Шаг 13: во вкладке General нажмите кнопку View Certificate. В появившемся окне Certificate выберите вкладку Details и нажмите кнопку Copy to File:

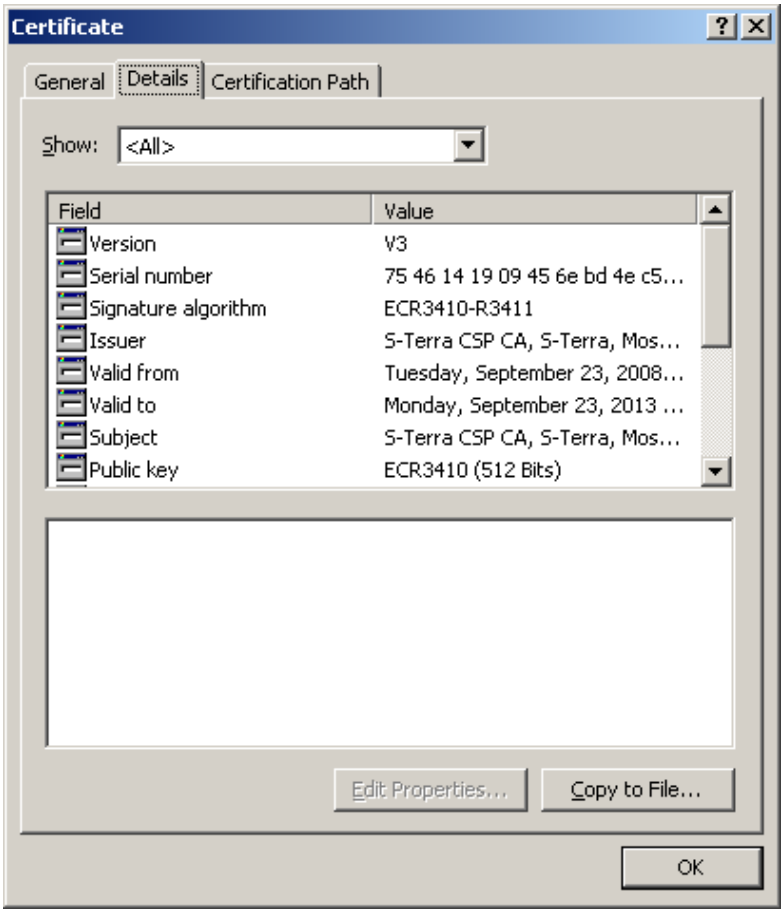

Рисунок 75

Шаг 14: далее в окне визарда `Certificate Export` (Рисунок 76) выберите формат, в котором должен быть экспортирован сертификат:

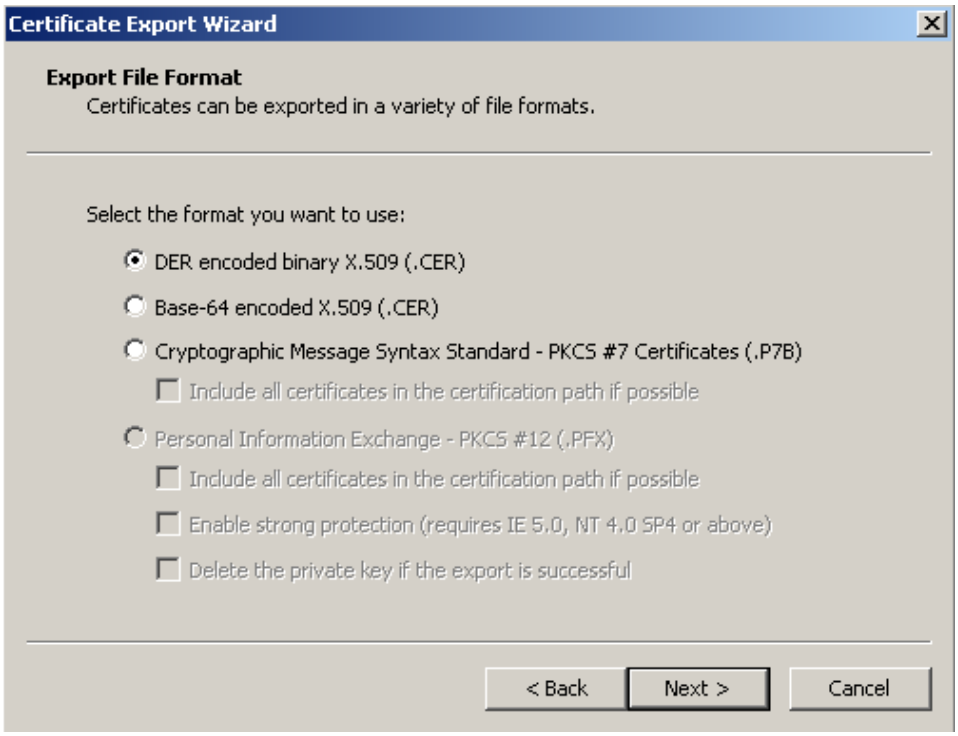

Рисунок 76

Шаг 15: введите имя файла, в который будет экспортирован сертификат:

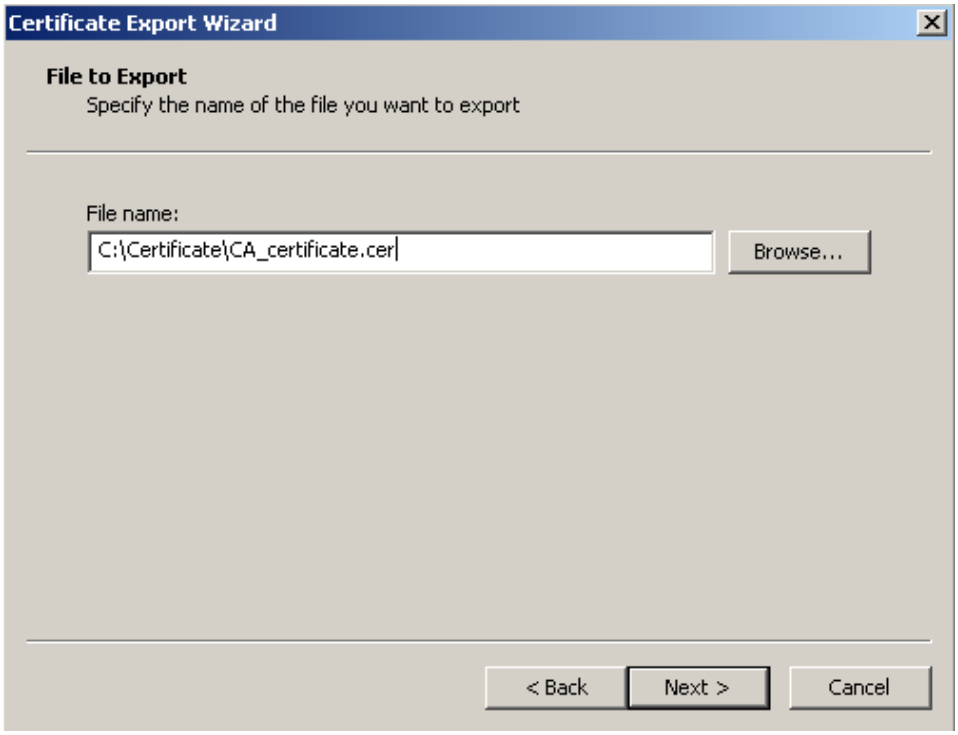

Рисунок 77

Шаг 16: экспортирование CA сертификата в файл завершено, нажмите `Finish`

Рисунок 78

Закройте окно `Cerificate` (Рисунок 75), нажав кнопку `OK`.

Шаг 17: для автоматического создания подписываемых сертификатов по запросу проведите некоторые настройки Удостоверяющего Центра: выберите центр сертификации, а затем Action - Properties. Далее в окне Properties во вкладке Policy Module нажмите кнопку Properties…В появившемся окне установите флажок Follow the settings in the certificate template, if applicable. Otherwise, automatically issue the certificate и нажмите OK (Рисунок 79).

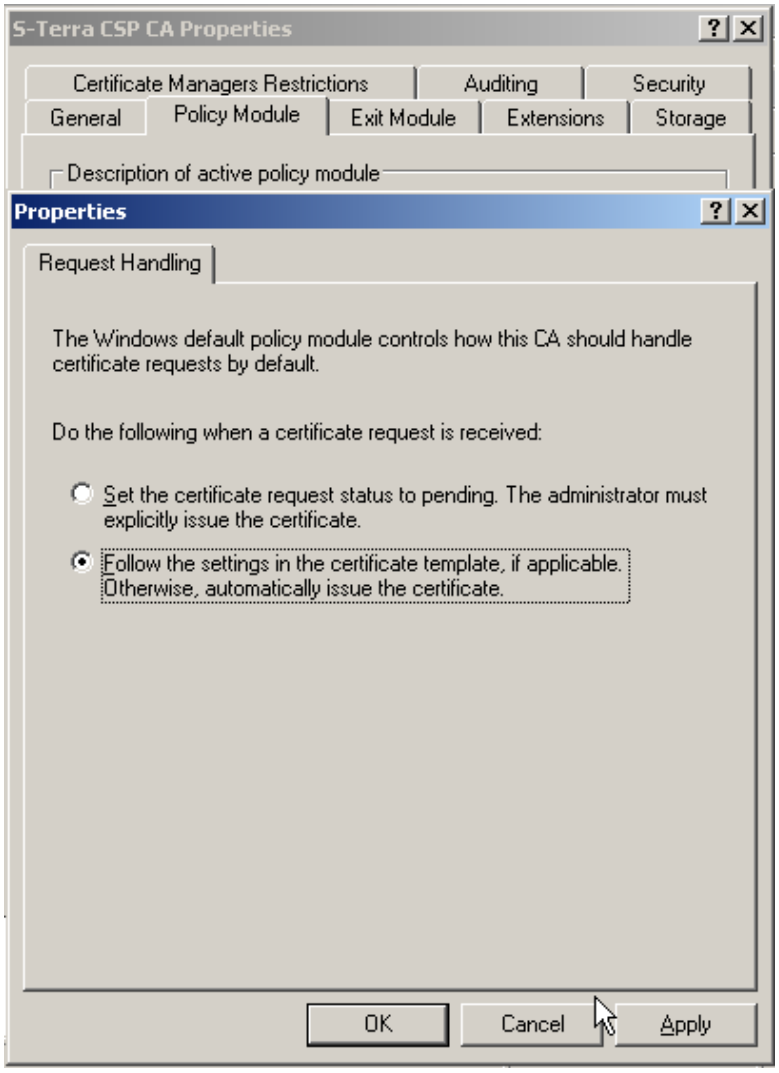

Рисунок 79

Появится предупреждение о необходимости перезапуска сервиса.

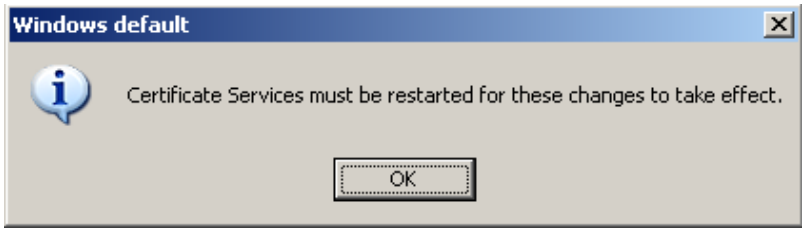

Рисунок 80

Шаг 18: в окне Certificate Authority выберите центр сертификации, затем меню `Action -- All Tasks – Stop Service`. После остановки сервиса выберите `Start Service`.

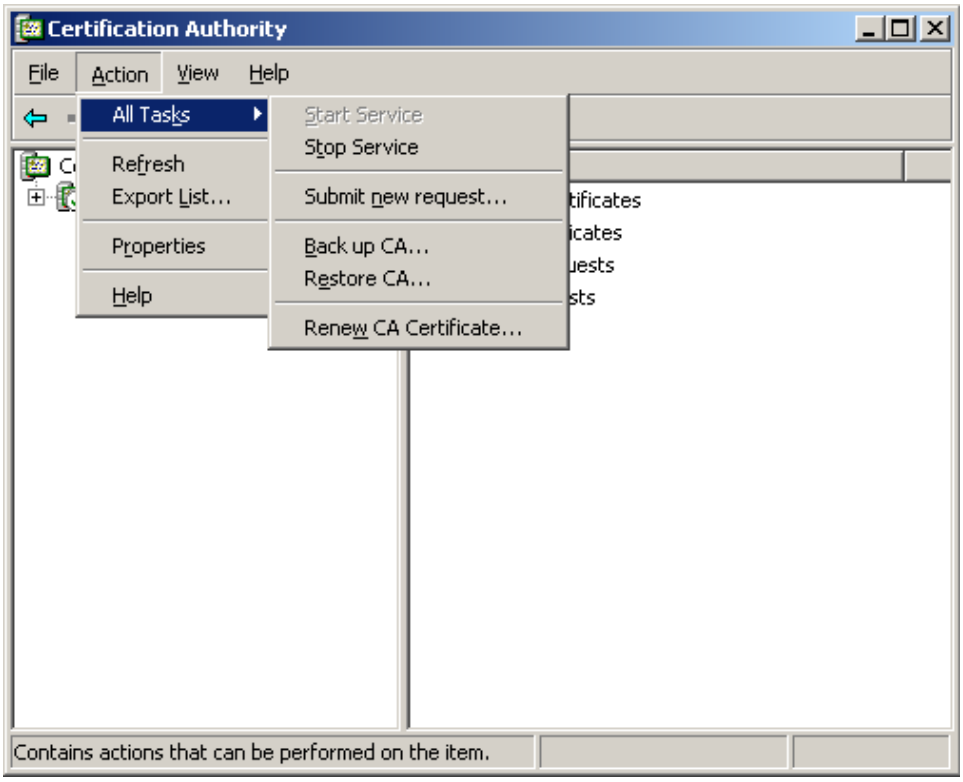

Рисунок 81

На этом создание Удостоверяющего Центра и его CA сертификата закончено.

Установка Admin-PKI

Для создания ключевой пары для сертификата и формирования запроса на локальный сертификат установите программный продукт Admin-PKI.

Для выполнения инсталляции необходимо:

- операционная система Windows 2000 Server (или выше). Возможно использовать тот же компьютер, на котором установлен Microsoft Certification Authority
- программный продукт Signal-COM Admin-PKI версии 3.1.0.4 (или выше)
- описание Admin-PKI можно посмотреть по адресу: http://www.signal-com.ru/ru/prod/pki/admin_pki/
- демо-версию этого продукта можно запросить по адресу: http://www.signal-com.ru/ru/demo/admin-pki/index.php

Запустите инсталляцию из файла `AdminPKI(trial).exe` и следуйте инструкциям инсталлятора:

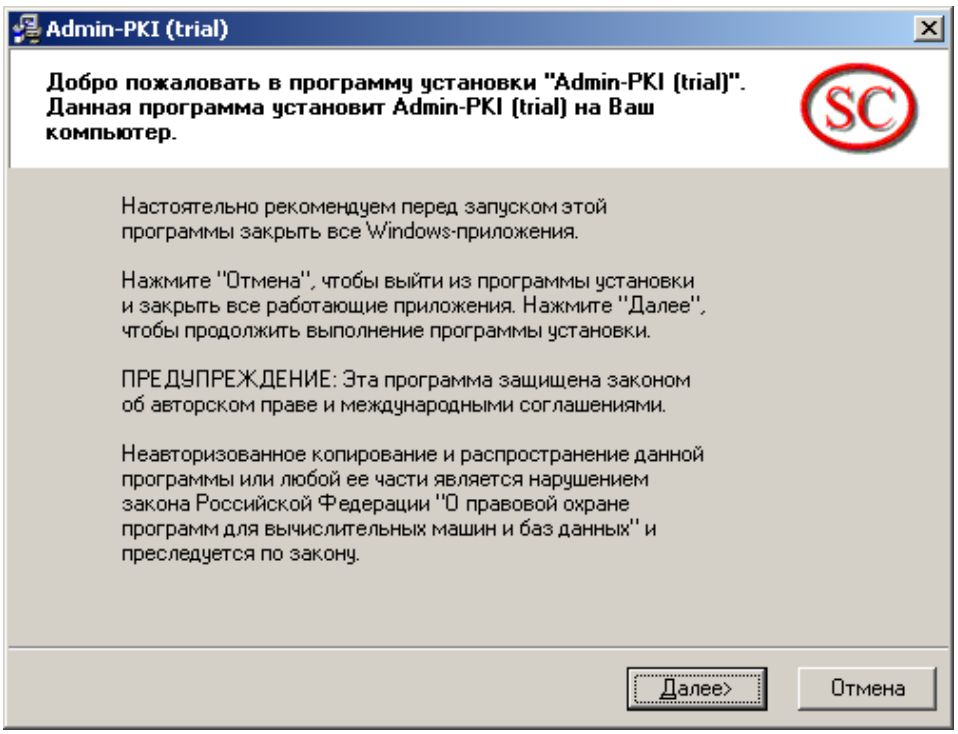

Рисунок 82

Далее везде нажимайте кнопку `Далее`:

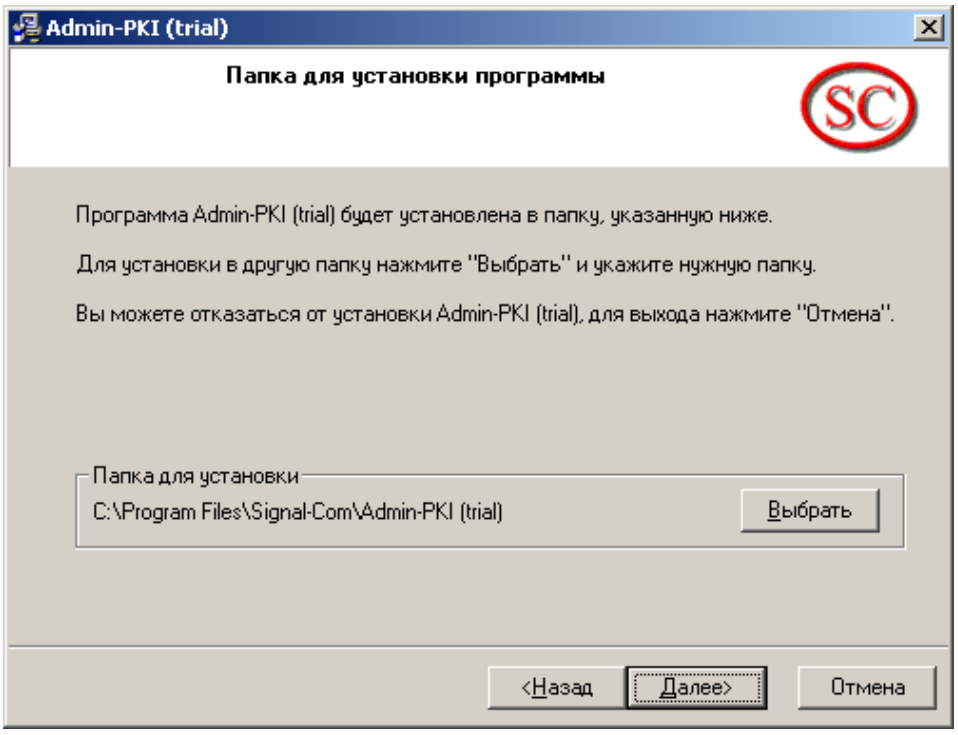

Рисунок 83

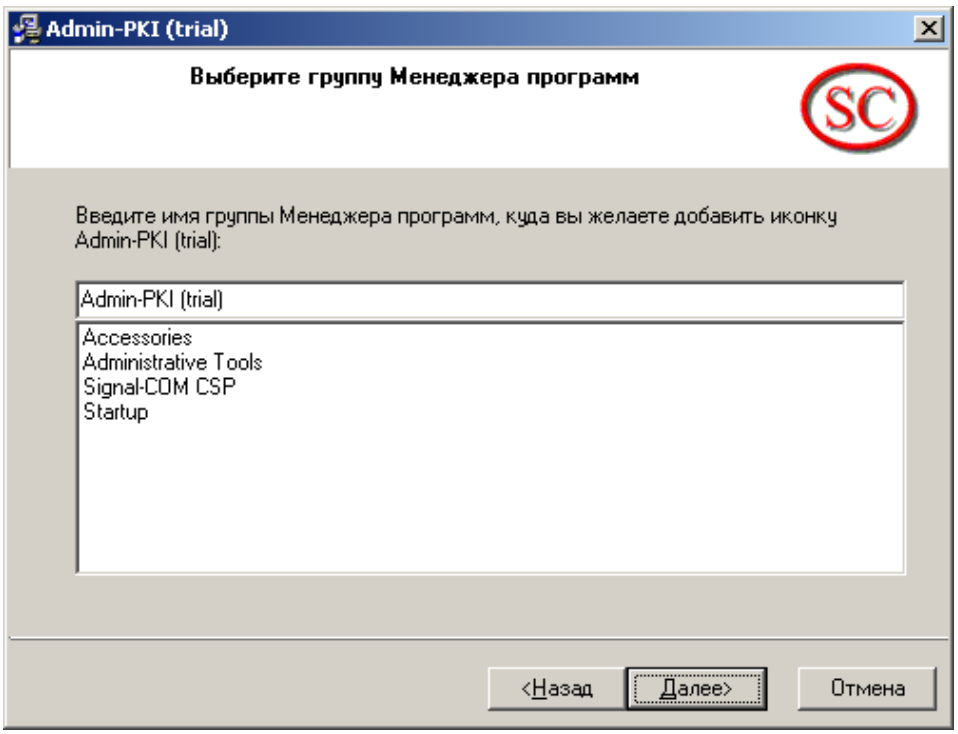

Рисунок 84

Инсталляция завершена, нажмите `Завершить`.

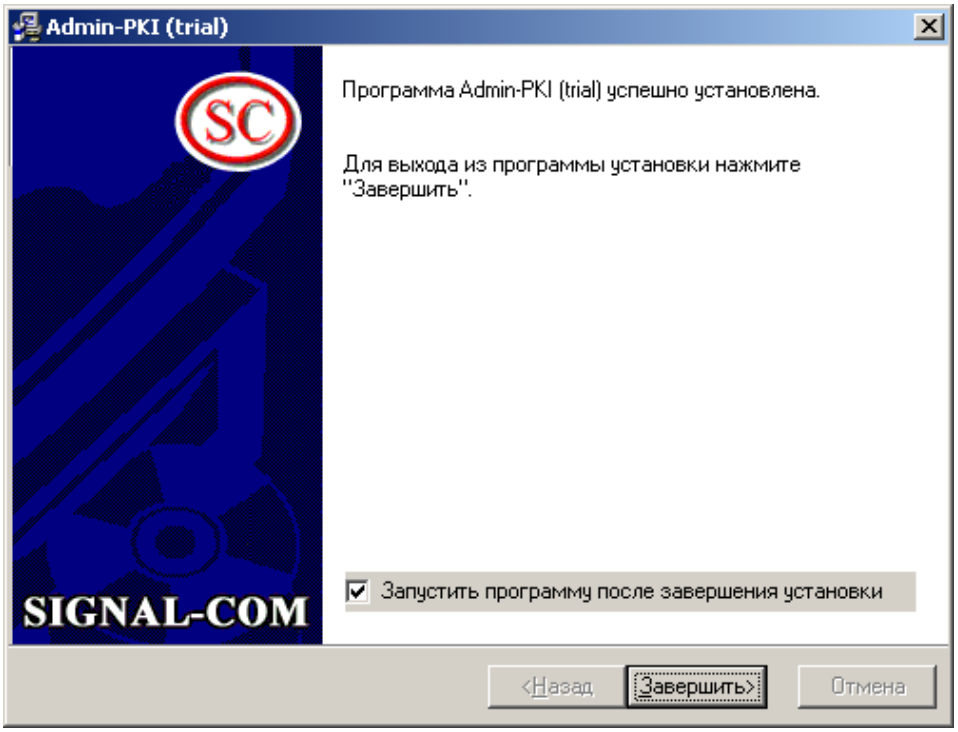

Рисунок 85

Создание ключевой пары и запроса на локальный сертификат с помощью Admin-PKI

Создание ключевой пары и запроса на локальный сертификат можно осуществить одним из двух способов – с помощью Admin-PKI или Приложения “KeyGen”.

При использовании приложения KeyGen на шлюзе безопасности для создания ключевой пары и запроса на сертификат, контейнер с секретным ключом локального сертификата размещается на шлюзе безопасности, остается доставить только CA и локальный сертификаты.

При использовании Admin-PKI – контейнер с секретным ключом размещается на жестком диске отдельного компьютера, который нужно доставить на шлюз безопасности, как и сертификаты.

Шаг 1: запустите Admin-PKI `(Start –Programs –Admin-PKI (trial)- Admin-PKI (trial) v3)`. В меню `Формирование` выберите пункт `Генерация ключей:`

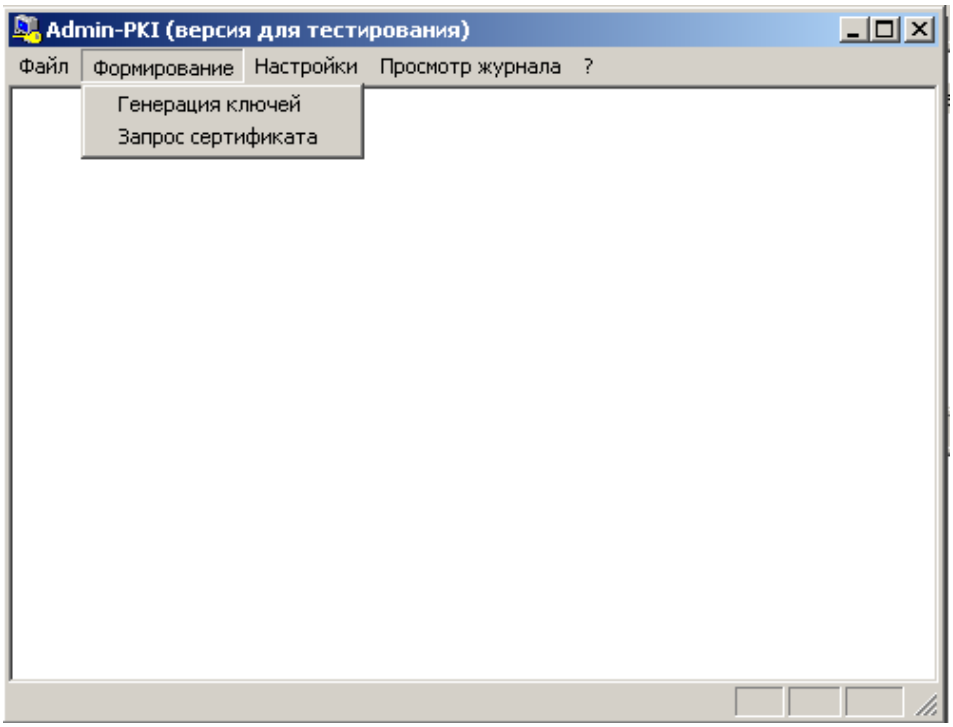

Рисунок 86

Шаг 2: в поле Каталог ключевого носителя задайте контейнер в виде каталога.

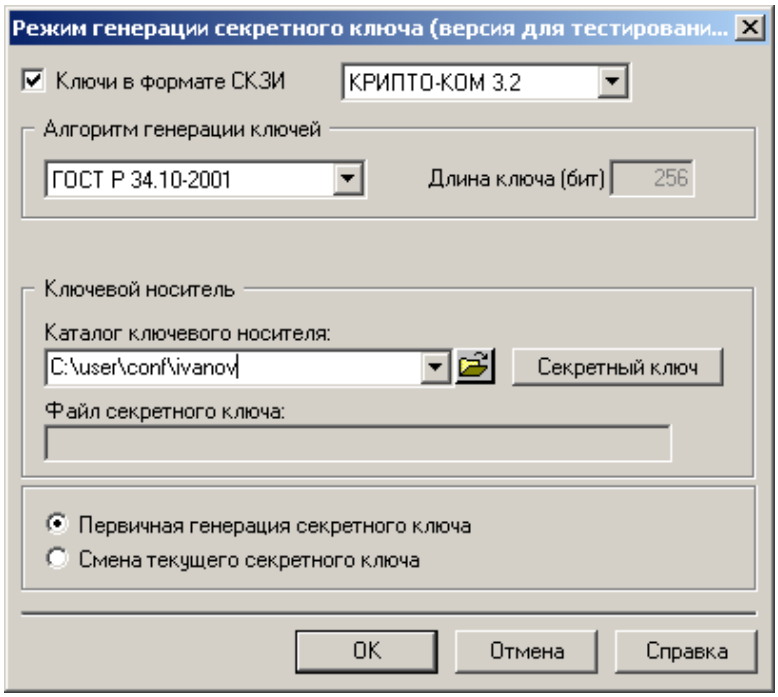

Рисунок 87

Шаг 3: если секретный ключ локального сертификата нужно разместить не в контейнере, а в отдельном файле, то в окне «`Режим генерации секретного ключа`» нажмите кнопку «`Секретный ключ`» и в окне «`Файл секретного ключа`» укажите имя секретного ключа и нажмите `OK`:

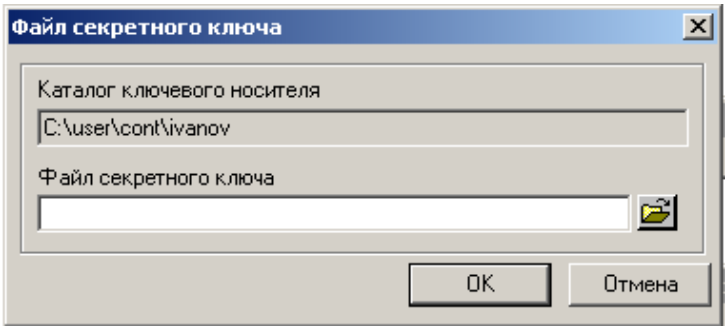

Рисунок 88

Шаг 4: на вопрос о задании нового каталога (контейнера) ответьте `Yes`:

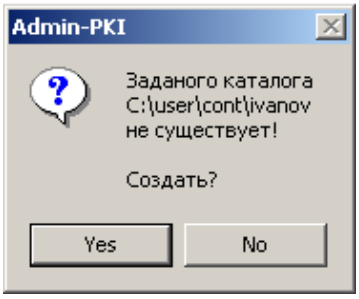

Рисунок 89

Шаг 5: ввести запрашиваемые параметры для локального сертификата и в поле «Файл запроса» указать имя файла, в который будет записан запрос на сертификат, и нажать OK:

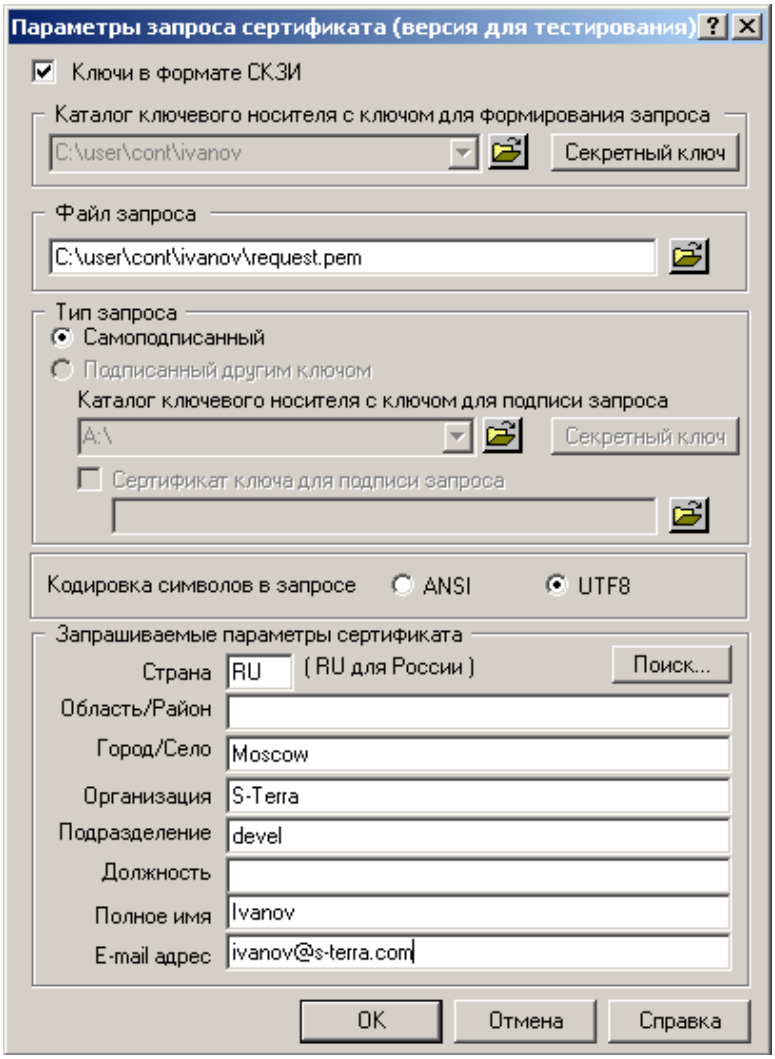

Рисунок 90

Примечание: если при создании запроса на сертификат при заполнении полей сертификата используются русские буквы, необходимо, чтобы они были введены в формате UTF-8.

Шаг 6: в окне отчета о выполненных операциях нажмите OK:

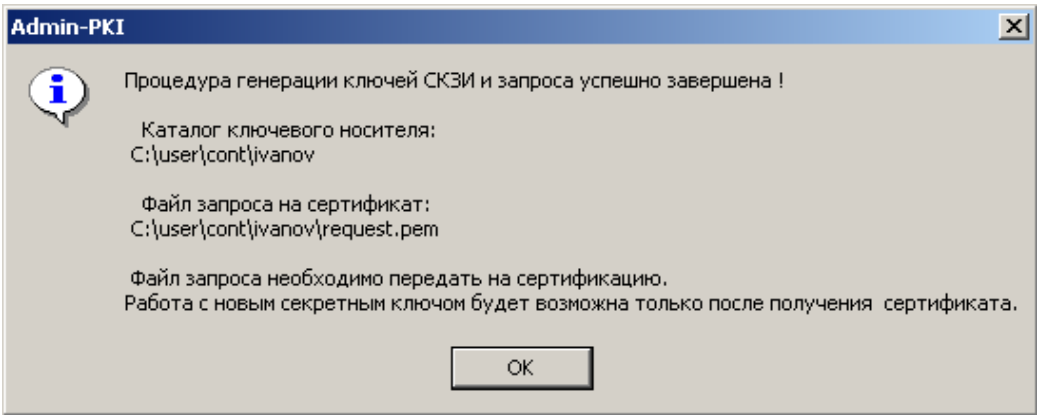

Рисунок 91

Шаг 7: в окне на вопрос о сохранении в буфере обмена запроса на сертификат нажмите `Yes`:

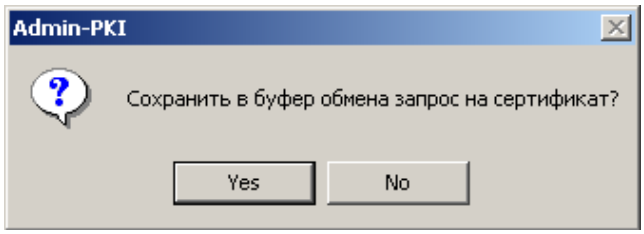

Рисунок 92

Шаг 8: на вопрос о передаче запроса на сертификат в Удостоверяющий Центр по E-mail нажмите `No`:

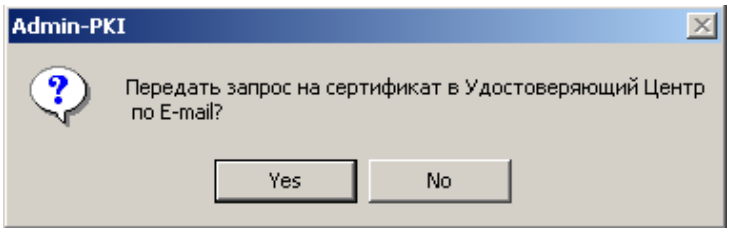

Рисунок 93

Шаг 9: на вопрос о передаче запроса на сертификат в Удостоверяющий Центр через Web-интерфейс Удостоверяющего Центра нажмите `No`:

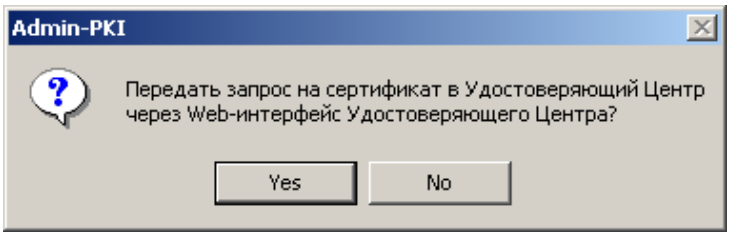

Рисунок 94

Шаг 10: на вопрос о распечатке запроса на сертификат нажмите `No`:

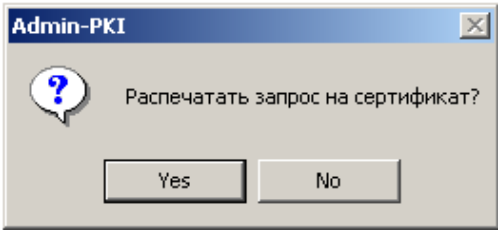

Рисунок 95

После этого Admin-PKI можно закрыть.

Контейнер с ключевой информацией и секретным ключом размещен в указанном Вами каталоге. А запрос на сертификат – в указанном файле. В данном примере:

- каталог контейнера – `C:\user\cont\ivanov`
- файл запроса на локальный сертификат – `C:\user\cont\ivanov\request.pem`.

Создание локального сертификата

Для создания локального сертификата нужно отослать запрос на сертификат в Удостоверяющий Центр, имеющий CA сертификат, созданный с использованием криптопровайдера Signal-COM CSP. В разделе "Установка и настройка Удостоверяющего Центра. Создание CA сертификата" описана установка и настройка такого УЦ Microsoft Certification Authority и создание для него CA сертификата.

Шаг 1: для отсылки запроса запустите Microsoft Internet Explorer и в поле для ввода URL с префиксом `http://` укажите IP-адрес Удостоверяющего Центра и утилиту `certsrv`, например, `http://10.0.34.33/certsrv`.

Шаг 2: в появившемся окне Удостоверяющего Центра выберите задачу –`Request a certificate` (Рисунок 96):

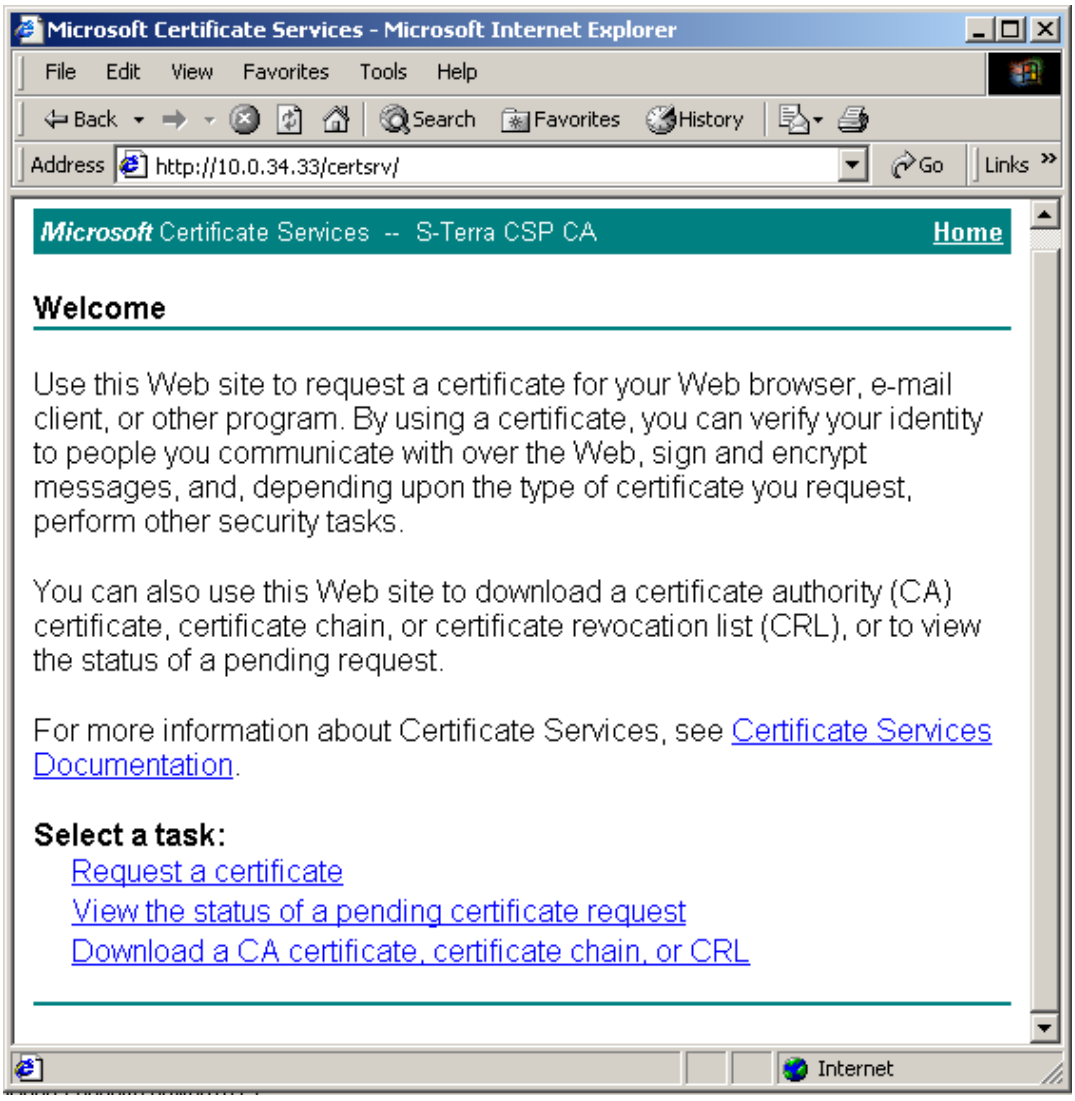

Рисунок 96

Шаг 3: выберите `advanced certificate request` (Рисунок 97):

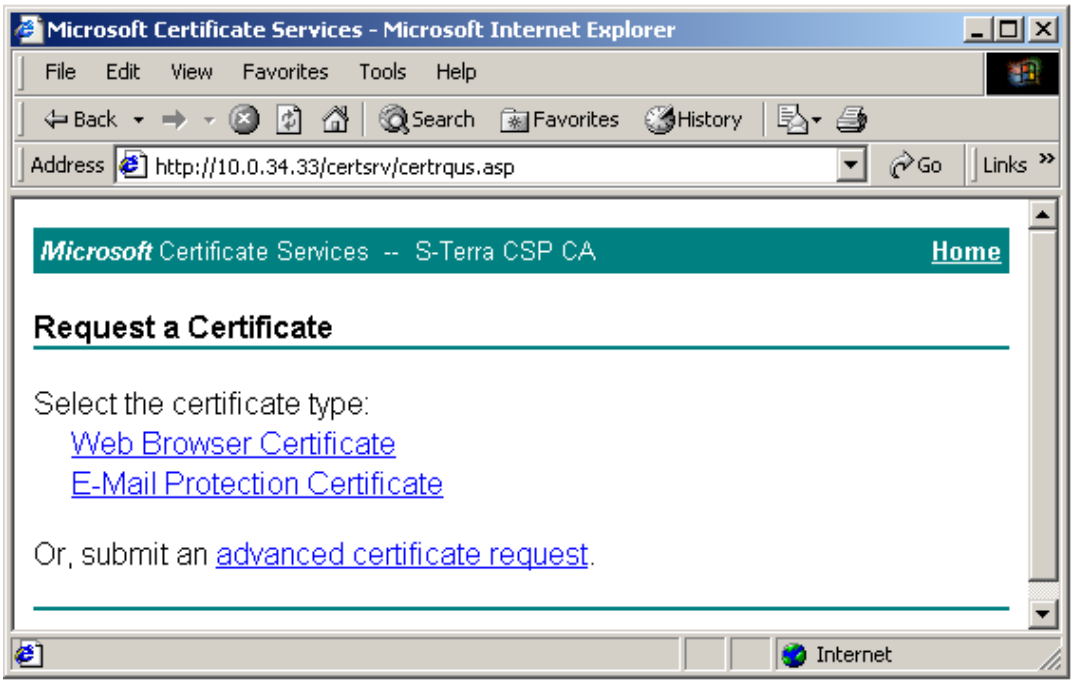

Рисунок 97

Шаг 4: выберите второе предложение `Submit a certificate request…..` (Рисунок 98):

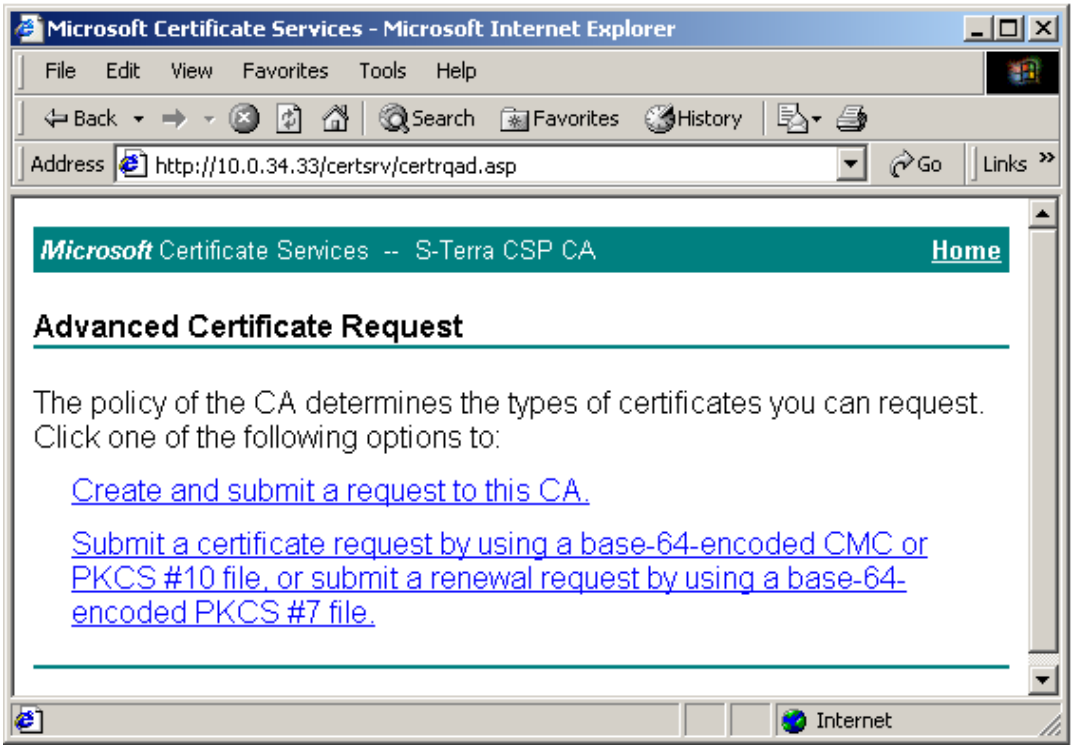

Рисунок 98

Шаг 5: скопируйте из файла запрос на сертификат, описанный и созданный в разделе "Создание ключевой пары и запроса на локальный сертификат", и вставьте его в поле `Saved Request` и нажмите `Submit` (Рисунок 99):

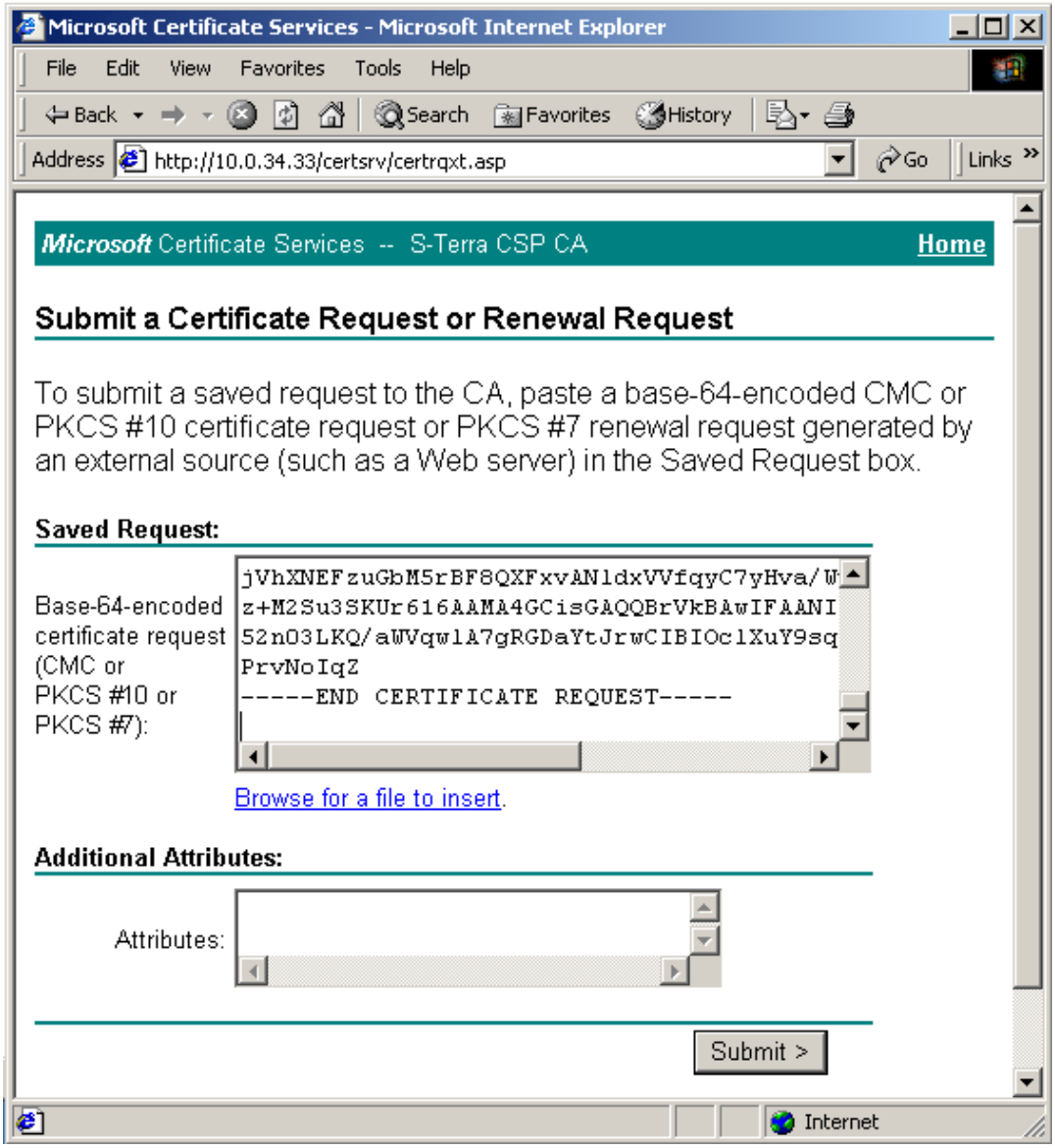

Рисунок 99

Шаг 6: Удостоверяющий Центр издал локальный сертификат по полученному запросу. Выберите формат файла сертификата и нажмите `Download certificate` (Рисунок 100):

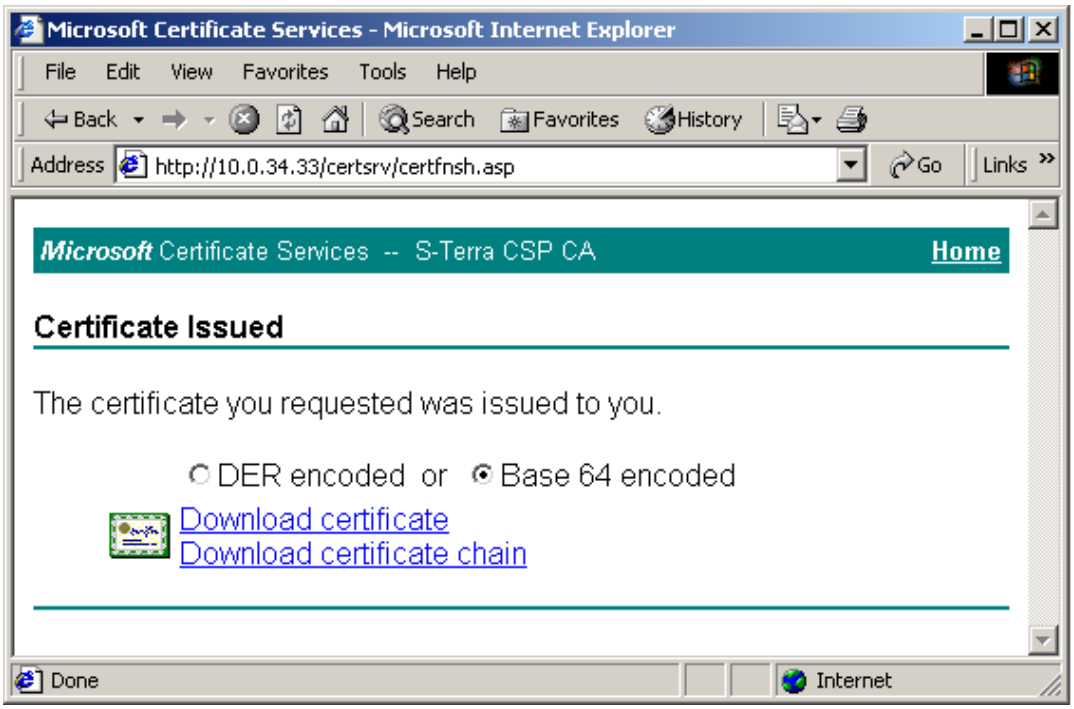

Рисунок 100

Шаг 7: установите переключатель в положение `Save this file to disk` и нажмите `OK` (Рисунок 101):

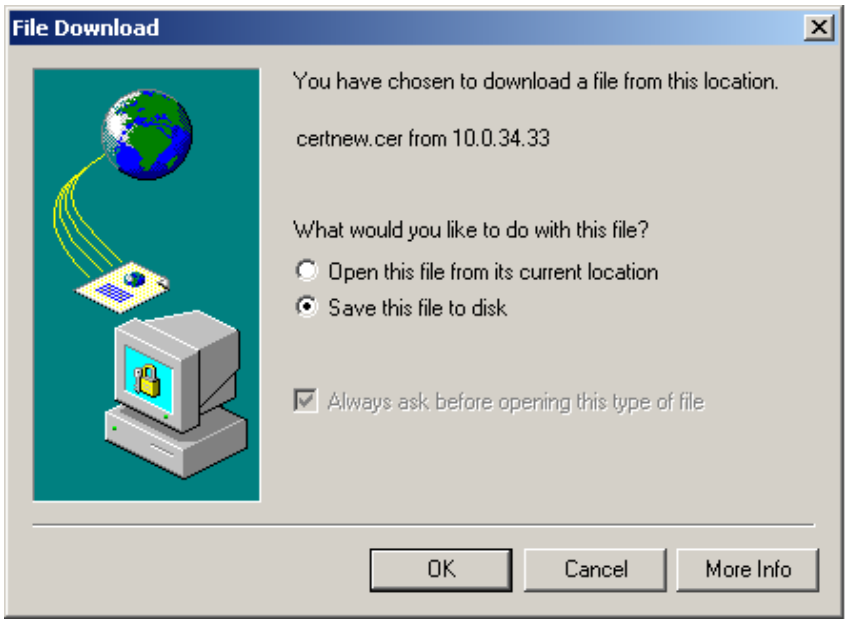

Рисунок 101

Шаг 8: введите имя файла, в который будет записан локальный сертификат и нажмите `Save`:

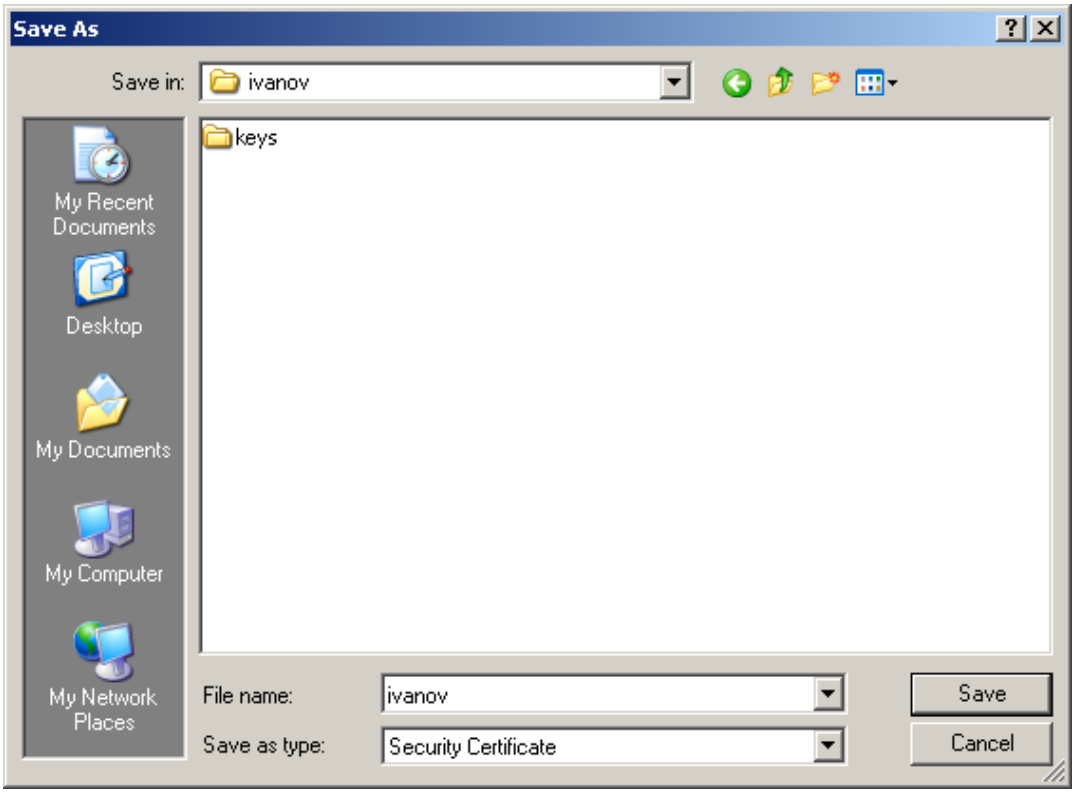

Рисунок 102

В результате всех выполненных действий мы создали CA сертификат и локальный сертификат, и экспортировали их в файлы. Эти сертификаты можно использовать при работе с продуктами CSP VPN Agent. Секретный ключ локального сертификата размещен в контейнере – `C:\user\cont\ivanov`. Для удобства и безопасности контейнер с секретным ключом лучше размещать на внешнем ключевом носителе.

Создание ключевой пары и запроса на локальный сертификат с использованием приложения "KeyGen"

Приложение "KeyGen", созданное компанией "Сигнал-КОМ", входит в состав дистрибутива CSP VPN Gate со встроенной криптобиблиотекой "Крипто-КОМ 3.2". На основе этого приложения можно создать ключевую пару и запрос на локальный сертификат в формате PKCS#10, а не только с помощью Admin-PKI.

Утилита `keygen` из состава приложения размещена в каталоге `/opt/Signal-COM/bin`.

Для создания ключевой пары и формирования запроса выполните следующие команды:

- `rm -rf pse_test`
- `mkdir pse_test`

- создайте текстовый конфигурационный файл с именем `keygen.conf`, в котором опишите параметры ключевого контейнера, параметры создаваемых ключей и атрибуты формируемого запроса на сертификат. Этот файл имеет примерно следующий вид:

```
<keygen>

<pse value="pse_test" />

<keys>
  <algorithm value="R3410" />
  <bits value="1024"/>
</keys>

<request>
  <DistinguishedName>
    <countryName value="RU"/>
    <stateOrProvinceName value="Zelenograd"/>
    <localityName value="Moscow"/>
    <organizationName value="S-Terra CSP"/>
    <organizationalUnitName value="QA"/>
    <title value="Test certificate"/>
    <emailAddress value="info@s-terra.com"/>
    <commonName value="pse_test_cert"/>
  </DistinguishedName>
</request>


</keygen>
```

- сформируйте ключевой контейнер

```
/opt/Signal-COM/bin/keygen -p -f keygen.conf
```

- создайте ключевую пару

```
/opt/Signal-COM/bin/keygen -k -f keygen.conf
```

- создайте запрос на локальный сертификат

```
/opt/Signal-COM/bin/keygen -r --req pse_test.req --outform=PEM
-f keygen.conf
```

Таким образом, на жестком диске создан ключевой контейнер в виде каталога `/pse_test` и запрос на сертификат в формате PKCS#10, записанный в файл `pse_test.req`.

При создании ключевой пары использован алгоритм ГОСТ Р 34.10-94 – строка `<algorithm value="R3410" />`.

Созданный запрос отсылается в Удостоверяющий Центр так, как это было описано в разделе "Создание локального сертификата", где по полученному запросу будет создан сертикат. Локальный сертификат нужно доставить любым способом на аппаратно-программный комплекс и утилитой `cert_mgr import` зарегистрировать сертификат в базе Продукта.

Совместная работа на разных криптопровайдерах

Для совместной работы, например, двух шлюзов безопасности, на одном из которых установлен CSP VPN Gate со встроенной криптобиблиотекой "Крипто-КОМ 3.2", разработанной компанией "Сигнал-КОМ", на втором - CSP VPN Gate с СКЗИ "КриптоПро CSP 3.6", необходимо:

- на первом шлюзе с помощью утилиты `keygen` создать ключевую пару и запрос на локальный сертификат, указав в строке `<algorithm value="ECR3410CP"/> (*` ГОСТ Р 34.10-2001 в формате «Крипто-Про»)
- созданный запрос отослать в Удостоверяюший Центр, имеющий CA сертификат, созданный с использованием криптопровайдера «КриптоПро CSP 3.6»
- для второго шлюза безопасности создать запрос с помощью криптопровайдера «КриптоПро CSP 3.6» и отослать его в УЦ с CA сертификатом этого же криптопровайдера

Созданные таким образом сертификаты с использованием двух криптопровайдеров являются совместимыми по российскому алгоритму ГОСТ Р 34.10-2001, используемому при формировании и проверке ЭЦП.

Совместимость работы двух шлюзов безопасности в режиме VKO (выработки общего секретного ключа) осуществляется в пределах одной криптобиблиотеки. Для шлюзов безопасности с криптобиблиотеками разных производителей такая совместимость в режиме VKO не тестировалась.

Продукты CSP VPN Gate с разными встроенными и внешними криптобиблиотеками могут взаимодействовать между собой не только при использовании западных, но и российских криптоалгоритмов.

