

**Программные комплексы  
"Шлюз безопасности  
CSP VPN Gate. Версия 3.0"  
и**

**"Шлюз безопасности  
CSP RVPN. Версия 3.0"**

**Приложение**

**Руководство  
администратора**

## Содержание

1. ОТЛИЧИТЕЛЬНЫЕ ОСОБЕННОСТИ РАБОТЫ ПРОДУКТА ПОД УПРАВЛЕНИЕМ ОС LINUX .....	4
2. ОТЛИЧИТЕЛЬНЫЕ ОСОБЕННОСТИ РАБОТЫ ПРОДУКТА ПОД УПРАВЛЕНИЕМ ОС SOLARIS 9 .....	5
3. ПЕРЕКЛЮЧЕНИЕ КОНСОЛИ НА ПОСЛЕДОВАТЕЛЬНЫЙ ПОРТ В ОС SOLARIS.....	6
4. НАСТРОЙКА NTP (NETWORK TIME PROTOCOL) .....	8
5. НАСТРОЙКА MTU НА МОДУЛЕ .....	10
6. СОЗДАНИЕ ВИРТУАЛЬНОГО ДИСКОВОДА (ОС SOLARIS).....	11
7. НАЧАЛЬНАЯ КОНФИГУРАЦИЯ ДЛЯ УДАЛЕННОГО УПРАВЛЕНИЯ ШЛЮЗОМ .....	14
8. ОПИСАНИЕ ФОРМАТА INI-ФАЙЛОВ .....	24
8.1. ФОРМАТ ДОПУСТИМЫХ СТРОК .....	24
9. КОНВЕРТОР .....	26
9.1. ОСНОВНАЯ ЛОГИКА РАБОТЫ .....	26
9.2. ОГРАНИЧЕНИЯ НА КОНВЕРТОР .....	27
9.3. ЛОГИКА ЗАПУСКА КОНВЕРТОРА.....	34
9.4. АЛГОРИТМ РАБОТЫ КОНВЕРТОРА.....	37
9.5. ВНУТРЕННИЕ НАСТРОЙКИ КОНСОЛИ И КОНВЕРТОРА .....	40
9.6. УПРАВЛЕНИЕ КОНВЕРТОРОМ С ПОМОЩЬЮ INI-ФАЙЛА .....	41
9.7. СООБЩЕНИЯ В ЛОГЕ ПРИ КОНВЕРТИРОВАНИИ .....	47
9.8. ОПИСАНИЕ ОБРАБОТКИ ИНТЕРФЕЙСОВ.....	50
9.9. ФОРМИРОВАНИЕ ИМЕН СТРУКТУР LSP ПРИ КОНВЕРТИРОВАНИИ.....	58
10. СОЗДАНИЕ ЛОКАЛЬНОГО СЕРТИФИКАТА С ИСПОЛЬЗОВАНИЕМ СКЗИ "КРИПТОПРО CSP" .....	61
10.1. Установка СКЗИ "КриптоПро CSP" .....	61
10.2. Настройка СКЗИ "КриптоПро CSP" .....	61
10.3. Подключение внешних ключевых считывателей (носителей) .....	62
10.4. Создание локального сертификата в "КриптоПро CSP 3.0" .....	62

10.4.1.Инсталляция ключевого носителя REGISTRY в "КриптоПро CSP 3.0" .....	62
10.4.2.Инсталляция считывателя и ключевого носителя EToken в "КриптоПро CSP 3.0" .....	69
10.4.3.Установка и настройка Удостоверяющего Центра. Создание CA сертификата.....	82
10.4.4.Создание ключевой пары и запроса на локальный сертификат с помощью MICROSOFT CERTIFICATE SERVICES .....	92
10.4.5.Экспортирование локального сертификата в файл.....	99
10.5. Создание локального сертификата в "КриптоПро CSP 2.0" .....	105
10.5.1.Инсталляция ключевого носителя REGISTRY в "КриптоПро CSP 2.0" .....	105
10.5.2.Инсталляция внешнего считывателя ключевой информации в "КриптоПро CSP 2.0" .....	108
10.5.3.Установка и настройка Удостоверяющего Центра. Создание CA сертификата.....	112
10.5.4.Создание ключевой пары и запроса на локальный сертификат .....	112
10.5.5.Создание локального сертификата с "КриптоПро CSP 2.0" ..	113
10.6. Создание ключевой пары и запроса на локальный сертификат с помощью утилиты CRYPTSP в ОС RED HAT LINUX 9 и ОС SOLARIS 9 ..	119
11. СОЗДАНИЕ ЛОКАЛЬНОГО СЕРТИФИКАТА С ИСПОЛЬЗОВАНИЕМ "SIGNAL-COM CSP" .....	120
11.1. Установка "SIGNAL-COM CSP" .....	120
11.2. Установка и настройка Удостоверяющего Центра. Создание CA сертификата .....	124
11.3. Установка ADMIN-PKI .....	133
11.4. Создание ключевой пары и запроса на локальный сертификат с помощью ADMIN-PKI.....	136
11.5. Создание локального сертификата .....	140
11.6. Создание ключевой пары и запроса на локальный сертификат с использованием приложения "KEYGEN" .....	145
12. СОЗДАНИЕ ЛОКАЛЬНОГО СЕРТИФИКАТА С ИСПОЛЬЗОВАНИЕМ СКЗИ "LIRSSL" .....	147
13. СОВМЕШТАЯ РАБОТА НА РАЗНЫХ КРИПТОПРОВАЙДЕРАХ .....	151
14. РАСШИРЕНИЯ СЕРТИФИКАТА (CERTIFICATE EXTENSIONS) .....	152

# 1. Отличительные особенности работы продукта под управлением ОС Linux

---

Отличительными особенностями продукта CSP VPN Gate 100/100В/100V от продуктов CSP VPN Gate 1000/3000/7000/10000 являются:

- продукт CSP VPN Gate 100/100В/100V работает под управлением операционной системы Linux
- для аппаратной платформы в качестве терминала можно использовать компьютер, подключенный к последовательному порту. Для подключения терминального компьютера к аппаратной платформе используется нуль-модемный кабель (5 проводов). На компьютере можно использовать терминальную программу, например, Windows HyperTerminal.

В терминальной программе HyperTerminal проведите настройки:

File-> Properties-> Settings-> Emulation-> VT100.

Во вкладке Connect To нажать на кнопку Configure и провести следующие настройки COM-порта:

```
Bits per second: 115200
Data bits: 8
Parity: None
Stop bits: 1
Flow control: None
```

Вход в операционную систему:

```
Имя пользователя - root
Пароль- отсутствует.
```

## 2. Отличительные особенности работы продукта под управлением ОС Solaris 9

Для некоторых аппаратных платформ, на которых предустановлен Продукт CSP VPN Gate 1000/3000/7000/10000, в качестве терминала можно использовать компьютер, подключенный к последовательному порту аппаратной платформы, по причине невозможности использовать монитор и клавиатуру, подключенные к разъемам аппаратной платформы.

Список таких аппаратных платформ следующий:

```
DELL PE 1950
DELL PE R200
HP DL140G3
HP DL360G5
HP DL360R05
HP DL365G1
```

Для подключения терминального компьютера к аппаратной платформе используется нуль-модемный кабель (5 проводов). На компьютере можно использовать терминальную программу, например, Windows HyperTerminal.

В терминальной программе HyperTerminal проведите настройки:

```
File-> Properties-> Settings-> Emulation-> VT100.
```

Во вкладке `Connect To` нажать на кнопку `Configure` и провести следующие настройки COM-порта:

```
Bits per second: 115200
Data bits: 8
Parity: None
Stop bits: 1
Flow control: None
```

Вход в операционную систему:

```
Имя пользователя - root
Пароль- отсутствует.
```



4. Если для консоли нужно использовать второй последовательный порт машины, то необходимо заменить в командах `ttya` на `ttyb`.

## 4. Настройка NTP (Network Time Protocol)

### OS Solaris

На шлюзе безопасности CSP VPN Gate в ОС Solaris для синхронизации часов с NTP-сервером точного времени, необходимо создать файл `/etc/inet/ntp.conf`, содержащий строку

```
server <server_addr>
```

где

`<server_addr>`      адрес NTP-сервера, который можно использовать для синхронизации.

После перезапуска ОС ntp-сервис будет запущен.

### ОС Linux

На шлюзе безопасности в ОС Linux для синхронизации часов с NTP-сервером точного времени необходимо выполнить следующие действия:

1. создайте файл `/etc/ntp.conf`, содержащий строки

```
server <server_addr>
restrict default ignore
```

где

`<server_addr>`      адрес NTP-сервера, который следует использовать для синхронизации.

Добавьте этот же адрес в файл `/etc/ntp/step-tickers`.

2. внешний NTP-сервер не должен изменять текущую конфигурацию или запрашивать ваш Linux NTP-сервер, поэтому введите следующую строку:

```
restrict <server_addr> mask 255.255.255.255 nomodify notrap noquery
```

3. если к вашему Linux NTP-серверу будут поступать запросы на NTP синхронизацию от других компьютеров вашей сети, то добавьте в файл строку:

```
restrict <addr_local_network> mask <addr_local_mask> nomodify
notrap
```

где

`<addr_local_network>`      адрес локальной подсети которую должен обслуживать Linux NTP-сервер

`<addr_local_mask>`      маска подсети.

4. для того, чтобы Linux NTP-сервер имел полный доступ к самому себе без любых ограничений, впишите строку:

```
restrict 127.0.0.1
```

5. сохраните полученный файл и для автоматического запуска NTP сервиса при каждом рестарте системы, выполните:

```
chkconfig ntpd on
```

Для первого запуска NTP сервиса без перезапуска системы выполните:

```
service ntpd start
```

Время при работе с сертификатами:

1. В сертификате время указано относительно Гринвича
2. Шлюз работает с сертификатами в локальном времени
3. Время жизни сертификата не зависит от временного пояса
4. Время жизни сертификата будет зависеть от сезонного перевода часов, т.к. время корректируется в фиксированный момент по локальному времени, поэтому может возникнуть сбой именно в момент перевода часов в разных поясах. Как только перевод будет окончен во всех поясах, время жизни сертификата в них будет одинаковым.

## 5. Настройка MTU на модуле

---

Для модуля настройка MTU сетевого интерфейса, который задает максимальный размер пакета, передаваемый без фрагментации, осуществляется в ОС Linux следующим образом:

в скрипте `/etc/init.d/sysconfig` в конце перед командой `exit` добавьте строчку

```
ifconfig ethX mtu YYYY
```

где `X` - номер интерфейса, а `YYYY` - размер MTU сетевого интерфейса.

Таким образом, устанавливается постоянное значение MTU.

Установка значения MTU интерфейса на время одной сессии (до перезагрузки ОС) в привилегированном режиме EXEC консоли осуществляется командой:

```
Router#run ifconfig eth0 mtu YYYY (для интерфейса fa 0/0)
```

```
Router#run ifconfig eth1 mtu YYYY (для интерфейса fa 0/1)
```

где `YYYY` - размер MTU сетевого интерфейса.

Если Вы войдете на модуль под пользователем `root`, то `run` использовать не нужно.

## 6. Создание виртуального дисковода (OC Solaris)

При работе с СКЗИ "КриптоПро CSP 2.0" секретные ключи размещаются в контейнерах, которые могут храниться на дискетах или других внешних ключевых носителях.

В связи с этим иногда возникают трудности, связанные с конструктивными особенностями некоторых аппаратных платформ – отсутствие флоппи-дисковода.

В данном разделе описан способ использования виртуального флоппи-дисковода и образа дискеты в качестве хранилища контейнеров, в случае отсутствия у аппаратной платформы флоппи-дисковода. Приведенные рекомендации прошли проверку для операционной системы – Solaris 8, версия КриптоПро CSP – 2.0 rev 2003.03.21.10.33

### Создание образа дискеты (DOS)

Образ дискеты можно сделать с помощью утилиты `dd.exe`, свободно распространяемой в интернете:

```
ftp://ftp.uu.net:/vendor/sun/solaris/x86/dd.exe
```

Утилита проста в использовании. Для создания образа дискеты нужно выполнить команду:

```
dd drive file
```

Например:

```
dd a: floppy.img
```

В этом случае образ дискеты с именем `floppy.img` будет создан в той же папке, в которой находится файл `dd.exe`.

### Создание образа дискеты (Solaris)

Для того, чтобы сделать образ дискеты в операционной системе Solaris выполним следующие шаги:

- Выполним команду

```
/etc/init.d/volmgt stop
```

- Вставим в дисковод дискету с контейнером
- Выполним команду

```
dd if=/dev/rdiskette0 of=floppy.img bs=512
```

В результате будет создан образ дискеты с именем `floppy.img`.

## Создание виртуального дисковода

Следующие шаги будем производить на компьютере, на котором будем использовать виртуальный дискковод:

- Разместим образ дискеты в файловой системе компьютера. Например, по пути `/opt/floppy.img`. Образ дискеты перенесем на этот компьютер по сети
- Удалим флоппи-дискковод из списка устройств, обслуживаемых Volume Management
- Перезапустим Volume Management
- Создадим каталог `/floppy/floppy0`.

**Пример:**

```
/etc/init.d/volmgt stop
sed -e 's/^use floppy/#use floppy/' /etc/vold.conf >
etc/vold.conf.tmp
mv /etc/vold.conf.tmp /etc/vold.conf
/etc/init.d/volmgt start
mkdir -p /floppy/floppy0
```

## Монтировка образа дискеты

Для монтировки/отключения образа дискеты предлагается использовать следующий скрипт (`floppy_container`):

```
#!/bin/sh
usage () {
    echo "usage: $0 umount|mount <floppy_image>" 1>&2
    exit 1
}
case $1 in
    mount) action=1;;
    umount) action=2;;
    *) usage;;
esac

img=$2

if [ $action -eq 1 ]; then
    if [ ! -f "$img" ]; then
        echo "$0: can't open '$img'" 1>&2
        exit 1
    fi
    dev=`lofiadm -a $img`
    mkdir -p /floppy/floppy0
    if mount -F pcfs $dev /floppy/floppy0; then
        exit 0
    else
        lofiadm -d $dev
        "$0: can't mount the image" 1>&2
        exit 1
    fi
fi

dev=`mount | awk '{if ( $1 == "/floppy/floppy0" ) print $3}'`
if [ "$dev" = "" ]; then
    echo "$0: image is not mounted" 1>&2
    exit 1
fi
```

```

if umount $dev; then
  lofiadm -d $dev
  exit 0
fi
echo "$0: can't unmount the image" 1>&2
exit 1

```

Для монтировки созданного нами образа дискеты выполним:

```
floppy_container mount /opt/floppy.img
```

Для того, чтобы отмонтировать образ дискеты, нужно будет выполнить:

```
floppy_container umount
```

## Автоматическая монтировка образа при загрузке ОС

Для автоматической монтировки образа дискеты при загрузке операционной системы предлагается использовать скрипт (fcontainer):

```

#!/bin/sh
[ -x /opt/floppy_container ] || exit 0
case $1 in
  start)
    /opt/floppy_container mount /opt/fcontainer.img
  ;;
  stop)
    /opt/floppy_container umount
  ;;
  *)
    echo "usage: fcontainer start|stop" 1>&2
    exit 1
  ;;
Esac

```

Выполним следующие действия:

- скрипт fcontainer расположим по пути

```
/etc/init.d/fcontainer
```

- скрипт floppy\_container расположим по пути

```
/opt/floppy_container
```

```
r.
```

- установим права на запуск:

```
chmod +x /opt/floppy_container
```

```
chmod +x /etc/init.d/fcontainer
```

- сделаем линки в соответствующие каталоги:

```
ln /etc/init.d/fcontainer /etc/rc2.d/S22fcontainer
```

```
ln /etc/init.d/fcontainer /etc/rc0.d/K91fcontainer.
```

## 7. Начальная конфигурация для удаленного управления шлюзом

Настройка шлюзов безопасности CSP VPN Gate и CSP RVPN производится одинаково, выполняют они одни и те же функции, поэтому в дальнейшем во всем документе будем использовать только одно наименование - CSP VPN Gate или Продукт.

Если планируется удаленно настраивать локальную политику безопасности шлюза при помощи консоли по протоколу SSH1, то после инсталляции CSP VPN Gate рекомендуется загрузить начальную конфигурацию, которая позволит в дальнейшем создать *защищенный канал* для настройки шлюза. При использовании протокола SSH2 загрузка начальной конфигурации не нужна.

Загрузка начальной конфигурации на шлюз безопасности должна осуществляться с локального терминала с помощью cisco-like консоли.

Для создания защищенного канала также необходимо на компьютер, с которого будет осуществляться удаленная настройка шлюза, установить CSP VPN Client с согласованной начальной конфигурацией для создания IPSEC SA между этим компьютером и шлюзом.

Ниже приведен пример настройки начальной конфигурации на шлюзе и удаленном компьютере.

Предположим, что на шлюзе безопасности один сетевой интерфейс FastEthernet0/1 с IP-адресом 192.168.13.1 (Рисунок 1) подключен к локальной сети и на нем установлен продукт CSP VPN Gate. К этой же локальной сети подключен компьютер с IP-адресом 192.168.13.2 для удаленной настройки шлюза и на нем установлен административный пакет CSP VPN Client AdminTool для создания инсталляционного пакета CSP VPN Client. Аутентификация сторон осуществляется при помощи предопределенного ключа со значением "adm123456".

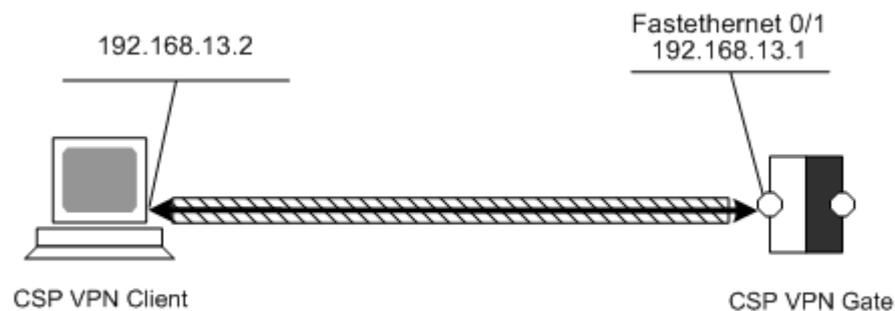


Рисунок 1

## Начальная конфигурация на шлюзе

Для создания начальной конфигурации на шлюзе безопасности запускаем cisco-like console командой `cs_console` из каталога `/opt/VPNagent/bin` (см. документ [«Cisco-like команды»](#)) и вводим следующие команды:

```
configure terminal
crypto isakmp policy 1
    hash md5
    encryption des
    authentication pre-share
    group 2
exit
crypto isakmp key adm123456 address 192.168.3.2
crypto ipsec transform-set adm esp-des esp-md5-hmac
exit
ip access-list extended adm
    permit tcp host 192.168.3.1 eq 22 host
192.168.3.2
    permit tcp host 192.168.3.1 eq 80 host
192.168.3.2
exit
crypto map adm 1 ipsec-isakmp
    match address adm
    set transform-set adm
    set pfs group2
    set peer 192.168.3.2
exit
interface FastEthernet0/1
    crypto map adm
exit
end
```

При выходе из конфигурационного режима политика безопасности будет загружена на шлюз безопасности.

## Начальная конфигурация на клиенте

На компьютере с установленным административным пакетом CSP VPN Client AdminTool (см. документацию «CSP VPN Client. Руководство администратора») запускаем графический интерфейс (Start – Programs – CSP VPN Client AdminTool –Package Maker) и создаем согласованную со шлюзом политику для создания защищенного соединения между ними.

Во вкладке Auth заполняем поля для настройки аутентификации на predeterminedном ключе "adm123456" и выбираем идентификатор.

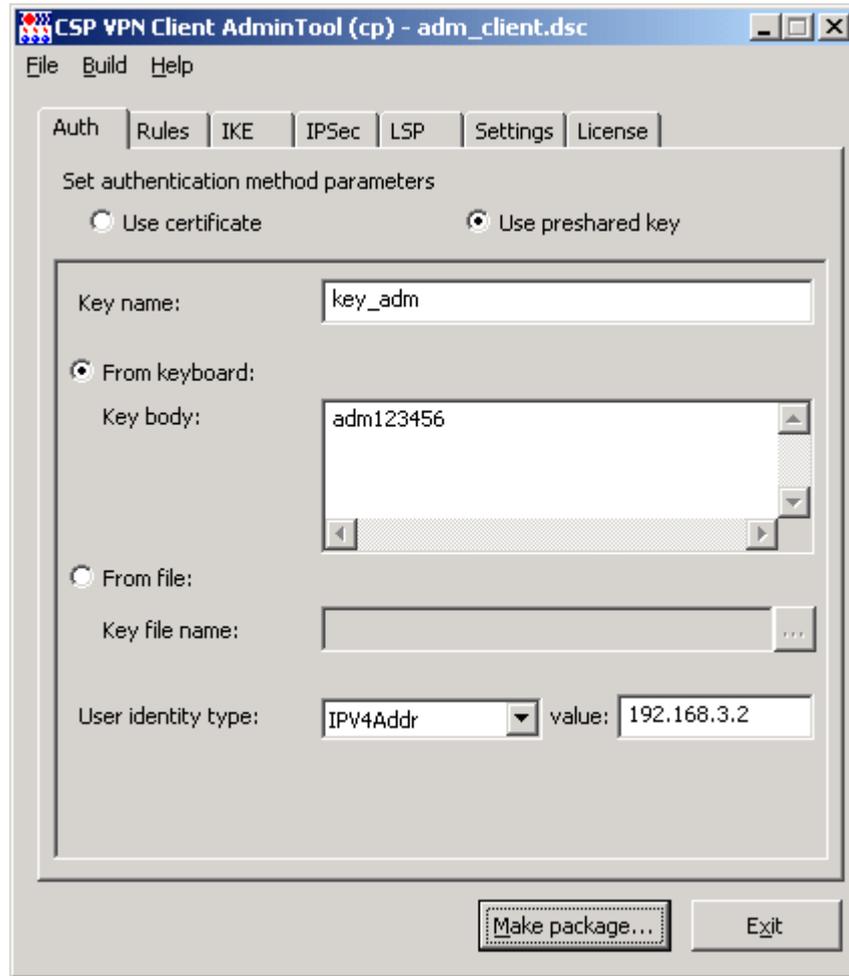


Рисунок 2

Во вкладке Rules создаем правило для создания защищенного соединения. Для этого нажимаем кнопку Add и устанавливаем параметры для правила (более подробно о работе с правилами см. «CSP VPN Client. Руководство администратора»).

В результате форма Add Rule должна иметь следующий вид (Рисунок 3):

**Add Rule**

Set rule parameters

**Local IP Addresses**

Any  
 Custom

IP Address	Subnet Mask
------------	-------------

Add... Edit... Remove

**Partner IP Addresses**

Any  
 Custom

IP Address	Subnet Mask
192.168.3.1	255.255.255.255

Add... Edit... Remove

**Services and Protocols**

Any  
 Custom

Name	Ports
SSH Client	-
HTTP Client	-

Add... Edit... Remove

**Action**

Pass  
 Drop  
 Protect using IPSec

Tunnel IP Addresses of IPSec partner:

Use random IP Address order

192.168.3.1	Up
-------------	----

Down

Add... Edit... Remove

OK Cancel

Рисунок 3

Нажимаем кнопку OK для сохранения правила.

Во вкладке Rules для выделенного нового правила нажимаем кнопку Up для повышения приоритета. Вкладка Rules будет иметь следующий вид:

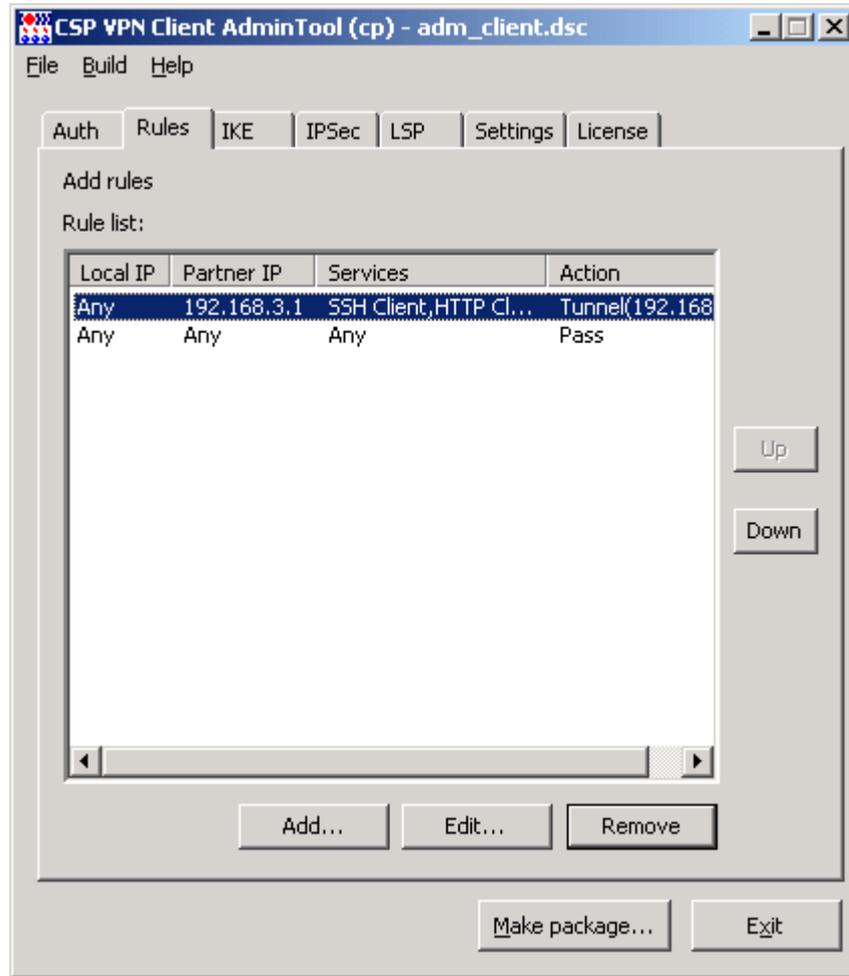


Рисунок 4

Во вкладке IPsec указываем значение группы MODP\_1024 для параметра Group:

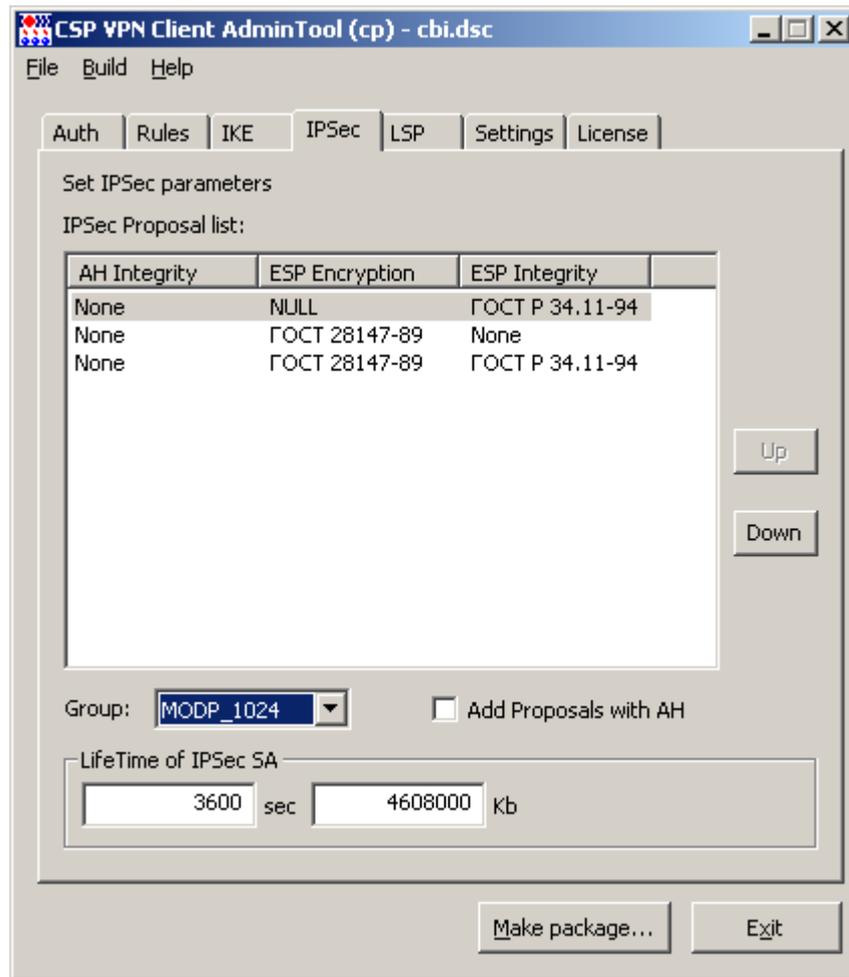


Рисунок 5

Переходим во вкладку LSP и нажимаем кнопку Refresh LSP. Созданная в предыдущих вкладках политика безопасности имеет следующий вид:

```
GlobalParameters (
  Title = "This LSP was automatically generated by CSP
VPN Client AdminTool (cp) at 2008.05.04 22:03:00"
  Version = "2.1"
  CRLHandlingMode = BEST_EFFORT
)
LDAPSettings (
  ResponseTimeout = 200
  HoldConnectTimeout = 60
  DropConnectTimeout = 5
)
IdentityEntry auth_identity_01 (
  IPv4Address *= 192.168.3.2
)
AuthMethodPreshared auth_method_01 (
  SharedIKESecret = "key_adm"
  LocalID = auth_identity_01
```

```
)
IKEParameters (
  DefaultPort = 500
  SendRetries = 5
  RetryTimeBase = 1
  RetryTimeMax = 30
  SACreationTimeMax = 60
  InitiatorSessionsMax = 30
  ResponderSessionsMax = 20
  BlacklogSessionsMax = 16
  BlacklogSessionsMin = 0
  BlacklogSilentSessions = 4
  BlacklogRelaxTime = 120
)
IKETransform ike_trf_01(
  LifetimeSeconds = 28800
  CipherAlg *= "G2814789CPR01-K256-CBC-65534"
  HashAlg *= "GR341194CPR01-65534"
  GroupID *= MODP_1536
)
IKETransform ike_trf_02(
  LifetimeSeconds = 28800
  CipherAlg *= "G2814789CPR01-K256-CBC-65534"
  HashAlg *= "GR341194CPR01-65534"
  GroupID *= MODP_1024
)
IKETransform ike_trf_03(
  LifetimeSeconds = 28800
  CipherAlg *= "G2814789CPR01-K256-CBC-65534"
  HashAlg *= "GR341194CPR01-65534"
  GroupID *= MODP_768
)
ESPTransform esp_trf_01(
  IntegrityAlg *= "GR341194CPR01-H96-HMAC-65534"
  CipherAlg *= "NULL"
  LifetimeSeconds = 3600
  LifetimeKilobytes = 4608000
)
ESPProposal esp_proposal_01(
  Transform *=esp_trf_01
)
ESPTransform esp_trf_02(
  CipherAlg *= "G2814789CPR01-K256-CBC-254"
  LifetimeSeconds = 3600
  LifetimeKilobytes = 4608000
)
ESPProposal esp_proposal_02(
  Transform *=esp_trf_02
)
ESPTransform esp_trf_03(
  IntegrityAlg *= "GR341194CPR01-H96-HMAC-65534"
```

```
CipherAlg *= "G2814789CPR01-K256-CBC-254"  
LifetimeSeconds = 3600  
LifetimeKilobytes = 4608000  
)  
ESPProposal esp_proposal_03(  
  Transform *=esp_trf_03  
)  
IKERule ike_rule(  
  DoNotUseDPD = FALSE  
  DPDIIdleDuration = 60  
  DPDResponseDuration = 5  
  DPDRetries = 3  
  MainModeAuthMethod *= auth_method_01  
  Transform *= ike_trf_01,ike_trf_02,ike_trf_03  
  IKECFGRequestAddress = TRUE  
  DoAutopass = TRUE  
)  
IPsecAction ipsec_action_01(  
  TunnelingParameters *=  
    TunnelEntry(  
      PeerIPAddress = 192.168.3.1  
    )  
  ContainedProposals *=  
(esp_proposal_01), (esp_proposal_02), (esp_proposal_03)  
  GroupID *= MODP_1024,MODP_1536,MODP_768  
  IKERule = ike_rule  
)  
FilterEntry local_entry_00_00(  
  ProtocolID *= 6  
)  
FilterEntry remote_entry_00_00(  
  IPAddress *= 192.168.3.1  
  ProtocolID *= 6  
  Port *= 80  
)  
FilteringRule filter_rule_00_00(  
  LocalIPFilter *= local_entry_00_00  
  PeerIPFilter *= remote_entry_00_00  
  Action *= (ipsec_action_01)  
  RefuseTCPPeerInit = FALSE  
)  
FilterEntry local_entry_00_01(  
  ProtocolID *= 6  
)  
FilterEntry remote_entry_00_01(  
  IPAddress *= 192.168.3.1  
  ProtocolID *= 6  
  Port *= 22  
)  
FilteringRule filter_rule_00_01(  
  LocalIPFilter *= local_entry_00_01
```

```
PeerIPFilter *= remote_entry_00_01
Action *= (ipsec_action_01)
RefuseTCPPeerInit = FALSE
)
FilterEntry local_entry_00_02(
  ProtocolID *= 17
)
FilterEntry remote_entry_00_02(
  IPAddress *= 192.168.3.1
  ProtocolID *= 17
  Port *= 22
)
FilteringRule filter_rule_00_02(
  LocalIPFilter *= local_entry_00_02
  PeerIPFilter *= remote_entry_00_02
  Action *= (ipsec_action_01)
  RefuseTCPPeerInit = FALSE
)
FilterEntry local_entry_01_00(
)
FilterEntry remote_entry_01_00(
)
FilteringRule filter_rule_01_00(
  LocalIPFilter *= local_entry_01_00
  PeerIPFilter *= remote_entry_01_00
  Action *= (PASS)
)
```

Во вкладке **Settings** укажите настройки протоколирования событий, а во вкладке **License** введите регистрационные данные на продукт CSP VPN Client с бланка Лицензии (см. «CSP VPN Client. Руководство администратора»).

После заполнения всех вкладок нажмите кнопку **Make package**, выберите тип инсталляции и сохраните инсталляционный файл на диске:

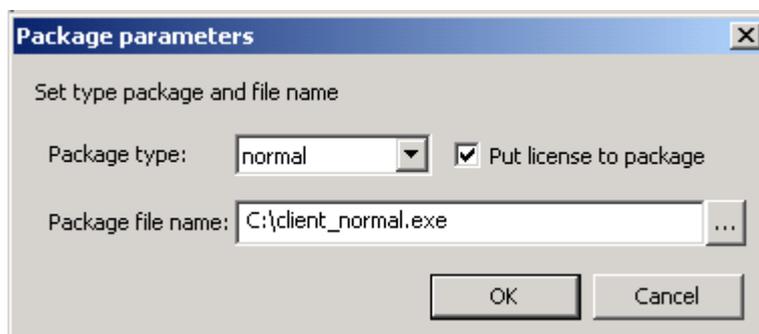


Рисунок 6

Установите на этом же компьютере CSP VPN Client, запустив созданный инсталляционный файл.

Таким образом, создана согласованная политика на шлюзе безопасности и клиенте, которая позволяет создавать *защищенный канал* для удаленной настройки шлюза при помощи консоли по протоколу SSH.

Дальнейшие действия по настройке шлюза описаны в документе [«Cisco-like команды»](#) или на странице “Поддержка” сайта компании в разделе “Типовые сценарии” по адресу <http://www.s-terra.com/CSP/RU/support/scn.htm>.

При изменении политики безопасности шлюза необходимо сохранять правило, созданное для его настройки.

## 8. Описание формата ini-файлов

### 8.1. 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.

## 9. Конвертор

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

### 9.1. 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 результате 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`.

## 9.2. 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 `crypto 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 реер будет возможна только в качестве ответчика. В качестве инициатора работа возможна только с первым реер.

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

#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 `permit`, 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 существовать 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 `crypto 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 сконvertированной 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.

## 9.3. 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
<p>Стартовая LSP загружена из <code>cs_converter</code>.</p>	<p>При входе в режиме <code>configure terminal</code> не происходит конвертирования.</p> <p>Инкрементальная настройка включена.</p> <p>Запуск конвертора осуществляется при выходе из режима <code>configure terminal</code> только в том случае, если были произведены какие-либо изменения в Cisco-like конфигурации.</p>	
<p>Стартовая LSP загружена из другого источника или не загружена вообще.</p> <p>Выдается сообщение в лог <a href="#">[1.7]</a>.</p>	<p>При входе в конфигурационный режим делается попытка сконвертировать текущую Cisco-like конфигурацию. Далее логика различается в зависимости от того, удалось сконвертировать конфигурацию или нет:</p> <p>Если удалось сконвертировать:</p> <ul style="list-style-type: none"> <li>инкрементальная настройка включена</li> <li>предыдущая агентская конфигурация (если она есть) сохраняется в файл <code>non_cscons.lsp</code> (расположение файла см. в <a href="#">Примечании</a>). Об этом выдается сообщение в лог <a href="#">[1.6]</a>. Если по каким-либо причинам не удалось сохранить файл, то сообщается в лог <a href="#">[3.11]</a>.</li> <li>запуск конвертора осуществляется при выходе из режима <code>configure terminal</code> только в том случае, если были произведены какие-либо изменения в Cisco-like конфигурации.</li> </ul> <p>Если не удалось сконвертировать, то выдается сообщение в лог <a href="#">[1.9]</a>. Инкрементальное конфигурирование <b>ВЫКЛЮЧЕНО</b>, поскольку нет LSP, к которой можно было бы корректно приложить инкрементальную LSP (политики рассогласованы). При выходе из режима <code>configure terminal</code> 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 <a href="#">[2.8]</a>).</p>	<p>При входе в режим <code>configure terminal</code> не происходит конвертирования.</p> <p>Инкрементальная настройка выключена. Выдается сообщение в лог <a href="#">[1.8]</a>.</p> <p>При выходе из режима <code>configure terminal</code> вызывается конвертор вне зависимости от того, были внесены изменения в Cisco-like конфигурацию или нет.</p> <p>Если конвертирование прошло успешно, то предыдущая конфигурация агента (если она есть) сохраняется в файл <code>non_cscons.lsp lsp</code> (расположение файла см. в <a href="#">Примечании</a>). Об этом выдается сообщение в лог <a href="#">[1.6]</a>. Если по каким-либо причинам не удалось сохранить файл, то сообщается в лог <a href="#">[3.11]</a>.</p>

**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").

## 9.4. 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 удастся 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 `presared 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`.

## 9.5. 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_mng` 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_mng` 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 [\[3.12\]](#).

## 9.6. 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`. Например: `ike-cipher-3des=DES3-K168-CBC`

- Редактирование пользователем 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 агента.
  - 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.
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):

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

[log]
severity          = NOTICE

[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              = GR341194CPR01-65534
ah-integrity-md5         = GR341194CPR01-H96-HMAC-254
esp-integrity-md5        = GR341194CPR01-H96-HMAC-65534
esp-cipher-des           = G2814789CPR01-K256-CBC-254
ike-cipher-des           = G2814789CPR01-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
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-3des          = DES3-K168-CBC
ike-cipher-aes           = AES-K128-CBC-7
ike-cipher-aes-192       = AES-K192-CBC-7
ike-cipher-aes-256       = AES-K256-CBC-7

[auth_cert]
send_request              = ALWAYS

[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              = GR341194CPR01-65534
ah-integrity-md5         = GR341194CPR01-H96-HMAC-254
esp-integrity-md5        = GR341194CPR01-H96-HMAC-65534
esp-cipher-des           = G2814789CPR01-K256-CBC-254
ike-cipher-des           = G2814789CPR01-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

[auth_cert]
send_request      = ALWAYS
send_cert         = ALWAYS
send_cert         = ALWAYS
```

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
```

```
[auth_cert]
send_request             = ALWAYS
send_cert               = ALWAYS
```

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           = GR341194CPR01-65534
ah-integrity-md5       = GR341194CPR01-H96-HMAC-254
esp-integrity-md5      = GR341194CPR01-H96-HMAC-65534
esp-cipher-des         = G2814789CPR01-K256-CBC-254
ike-cipher-des         = G2814789CPR01-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

[auth_cert]
send_request         = ALWAYS
send_cert            = ALWAYS
```

## 9.7. 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	File "{1}" opened for writing	Файл {1} открыт для записи конфигурации
1.6	Previous user-defined LSP saved in file "{1}"	Предыдущая пользовательская LSP сохранена в файле "{1}"
1.7	Non-synchronized policy detected. Policy type: <type> где <type> один из: DDP Drop All User-defined (Source: <source>), где <source> – Agent или Command-line utility.	Обнаружена несинхронизированная политика. Тип политики: <type>
1.8	Incremental policy loading disabled by policy_sync setting (file cs_conv.ini)	Инкрементальная политика отключена из-за настройки policy_sync (файл cs_conv.ini)
1.9	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.

	Сообщение	Комментарий
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 реер-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 реерс с разными preshared keys. Это не рекомендуемая ситуация.  Подробнее см. <a href="#">п.8 для несовпадающих Preshared keys</a> .

### 3. Ошибки

	Сообщение	Комментарий
3.1	Cannot read settings form INI file. Conversion failed.	Невозможно прочитать настройки из INI-файла.
3.2	LSP file "{1}" open for write failed.	Не удастся открыть файл {1} на запись агентской LSP
3.3	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.
3.4	No interfaces were found in the configuration.	В импортируемой конфигурации не заданы интерфейсы. Конвертирование не имеет смысла.
3.5	Interface "{1}" not found in the INI file. Conversion aborted.	Интерфейс {1} не задан в INI-файле. Конвертирование остановлено.
3.6	Certificate parse failed	Не удалось разобрать введенный сертификат.

	Сообщение	Комментарий
3.7	<p>Could not convert crypto map "{1}". Reason: &lt;Reason&gt;</p> <p>где &lt;Reason&gt;:</p> <ul style="list-style-type: none"> <li>There is no isakmp policy.</li> <li>There is no CA or appropriate preshared key. Also isakmp policy can have wrong type (rsa-sig or pre-share).</li> <li>There is no peer.</li> <li>There are no transform sets.</li> <li>Crypto map is incomplete.</li> <li>Unknown.</li> </ul>	<p>Невозможно сконвертировать crypto map "{1}". Причина: &lt;Причина&gt;</p> <p>где &lt;Причина&gt; одна из:</p> <ul style="list-style-type: none"> <li>Отсутствует isakmp policy.</li> <li>Отсутствует CA или подходящий Preshared Key, либо isakmp policy неправильного типа (rsa-sig или pre-share).</li> <li>Отсутствует peer.</li> <li>Отсутствуют transform sets</li> <li>Crypto map неполная (не хватает crypto ACL, transform set или peer).</li> <li>Неизвестная причина.</li> </ul>
3.8	LSP load failed	Не удалось загрузить сформированную LSP
3.9	Unsupported network wildcard "{1}"	Не поддерживается данный формат маски подсети
3.10	LSP conversion failed	Произошла некоторая невыясненная ошибка
3.11	Could not save previous user-defined LSP in file "{1}"	Не удалось сохранить предыдущую пользовательскую LSP в файл {1}
3.12	Could not save internal settings in file "{1}"	Не удалось сохранить внутренние настройки в файл {1}

## 9.8. Описание обработки интерфейсов

- 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
```

- 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 )
)
# ...
```

## 9.9. 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

## 10. 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.

### 10.1. Установка СКЗИ "КриптоПро CSP"

На отдельном компьютере с ОС Microsoft Windows установите СКЗИ "КриптоПро CSP 2.0 или 3.0". Процедура инсталляции описана в файле Installation.pdf, размещенном в папке CryptoPro\DOC дистрибутивного диска "КриптоПро CSP". После завершения процедуры инсталляции необходимо зарегистрировать "КриптоПро CSP 2.0", введя серийный номер и ключ активации с бланка Лицензии. Для "КриптоПро CSP 3.0" вводится только серийный номер.

### 10.2. 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 пользователя
- либо на локальном ключевом носителе (Registry) на компьютере пользователя.

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

Если контейнер с секретным ключом локального сертификата будет размещен в Registry, то этот носитель сначала нужно инсталлировать. Такая инсталляция описана в разделах ["Инсталляция ключевого носителя Registry в "КриптоПро CSP 2.0"](#) и ["Инсталляция ключевого носителя Registry в "КриптоПро CSP 3.0"](#).

Если контейнер расположен на внешнем ключевом носителе, то сначала нужно подключить к компьютеру считыватель ключевой информации, а затем инсталлировать его.. Подключение внешних считывателей ключевой информации, например e-Token и др. описано в разделе ["Подключение внешних ключевых считывателей"](#).

После установки "КриптоПро CSP" сразу же инсталлирован только один считыватель – дисковод a:\. В "КриптоПро CSP" 2.0 после подключения считывателя достаточно только его инсталлировать. В "КриптоПро CSP" 3.0 нужно инсталлировать не только внешний считыватель, но и его носитель информации. Процедура инсталляции внешних считывателей описана в разделе ["Инсталляция внешнего считывателя ключевой информации в "КриптоПро CSP 2.0"](#) и ["Инсталляция считывателя и ключевого носителя eToken в "КриптоПро CSP 3.0"](#).

## 10.3. 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" не входят драйвера, обеспечивающие взаимодействие "КриптоПро CSP" с внешними ключевыми носителями. С Web-страницы компании Крипто-ПРО

<http://www.cryptopro.ru/CryptoPro/products/csp/readers.htm> загрузите и установите модуль поддержки внешнего считывателя СКЗИ "КриптоПро CSP". Модуль поддержки eToken для СКЗИ "КриптоПро CSP 3.0" можно загрузить со страницы <http://www.aladdin.ru/support/download/category254>.

## 10.4. Создание локального сертификата в "КриптоПро CSP 3.0"

### 10.4.1. Инсталляция ключевого носителя Registry в "КриптоПро CSP 3.0"

Для инсталляции локального ключевого носителя Registry надо выполнить следующие действия:

**Шаг1** : запустить КриптоПро CSP: Start –Settings-Control Panel – CryptoPro CSP

**Шаг2** : в появившемся окне Properties войти во вкладку Hardware и нажать кнопку Configure readers (настроить считыватели) (Рисунок 7):

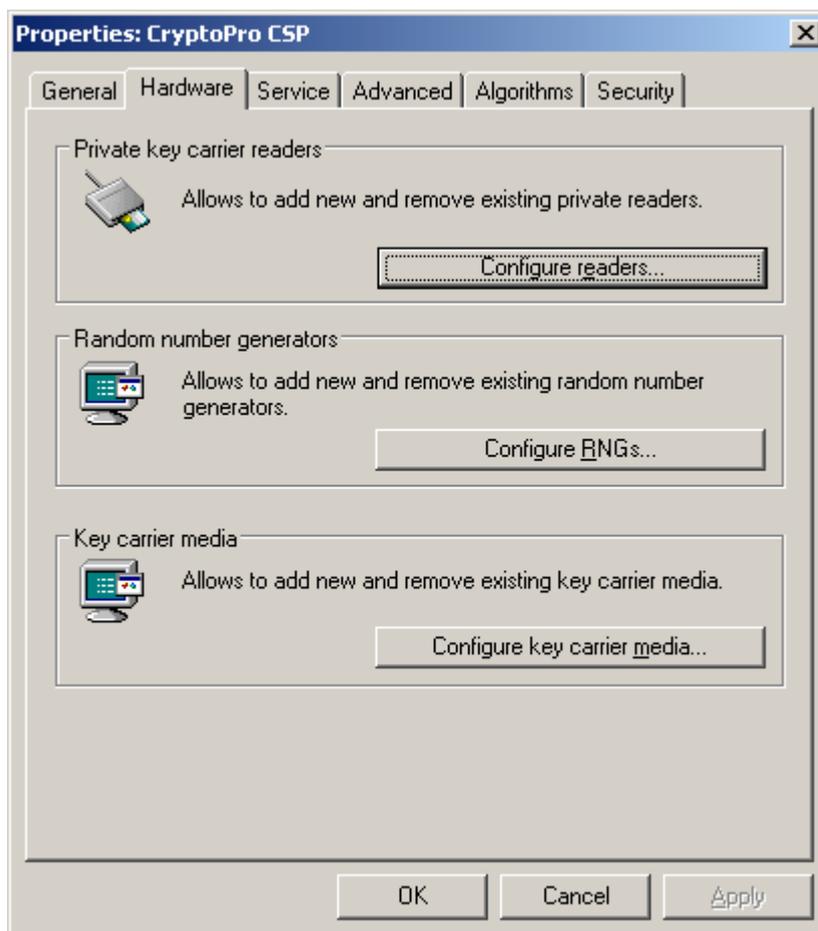


Рисунок 7



Рисунок 8

Шаг3 : нажать кнопку Add, чтобы добавить новый ключевой носитель (Рисунок 8).

Шаг4 : в окне визарда для инсталляции считывателя нажать кнопку Next:

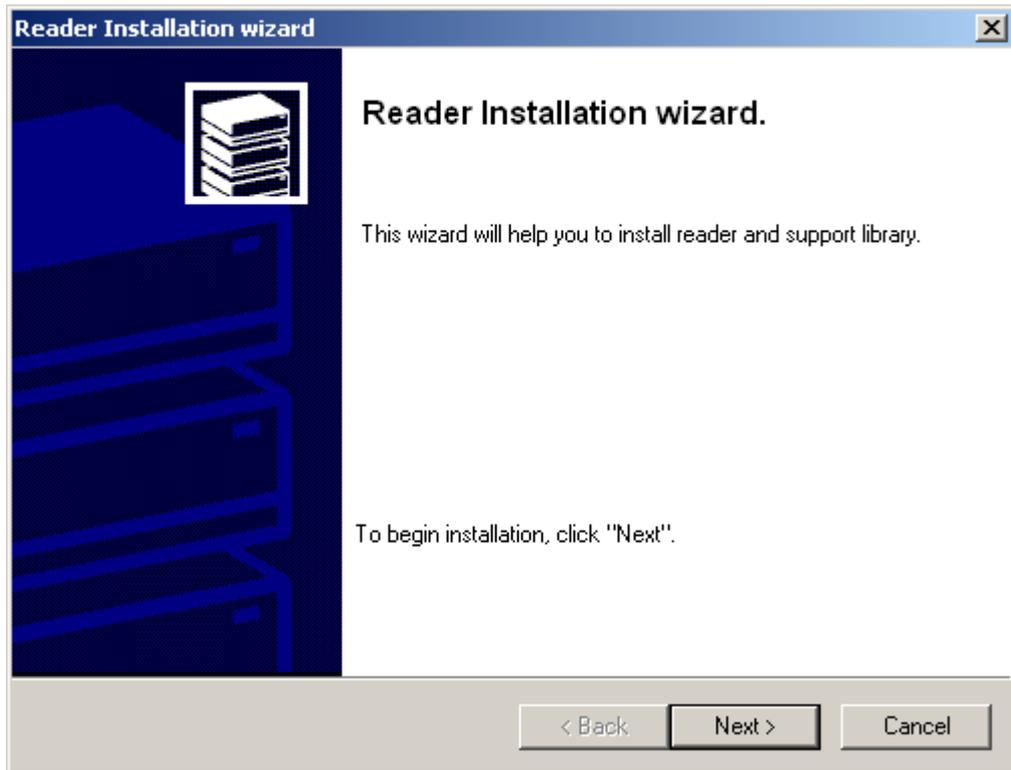


Рисунок 9

Шаг5 : для выбора инсталлятора считывателя нажмите кнопку Have disk...

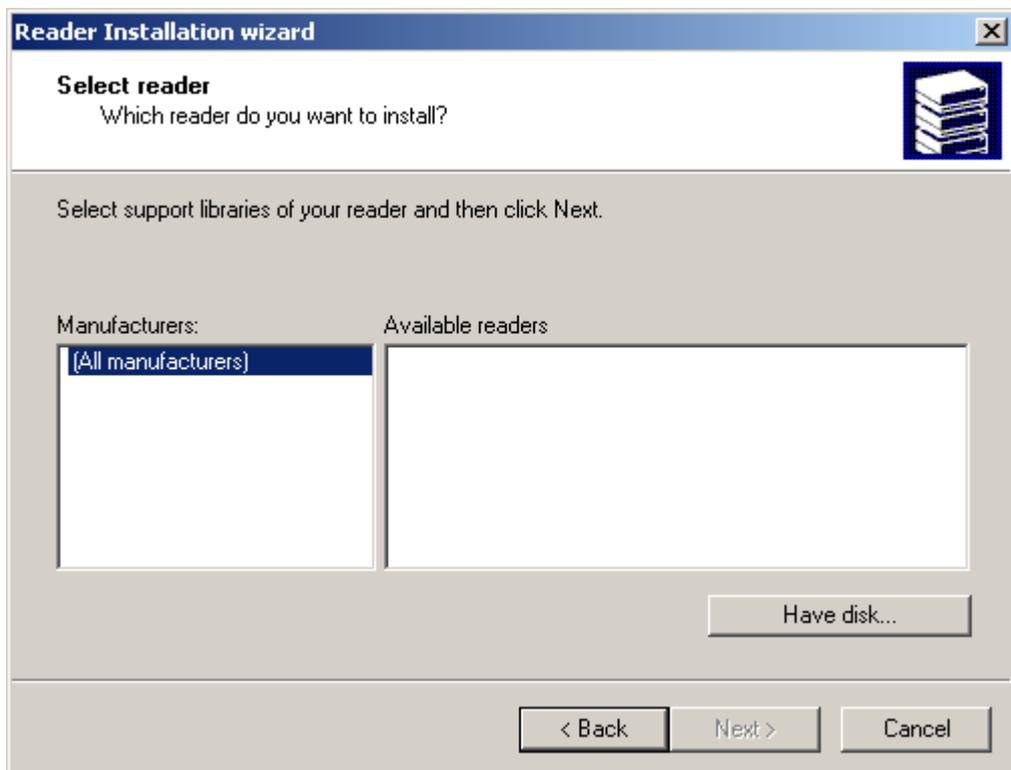


Рисунок 10

Шаг 6: и далее нажмите Next:

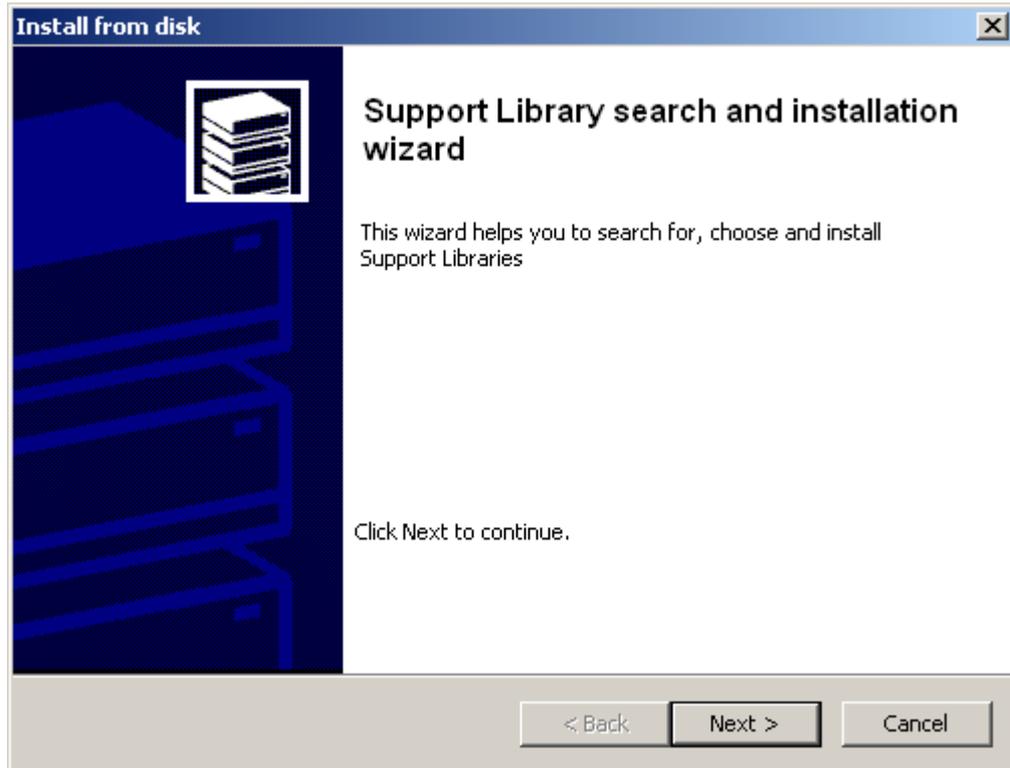


Рисунок 11

Шаг 7: укажите местоположение инсталлятора считывателя и нажмите Next:

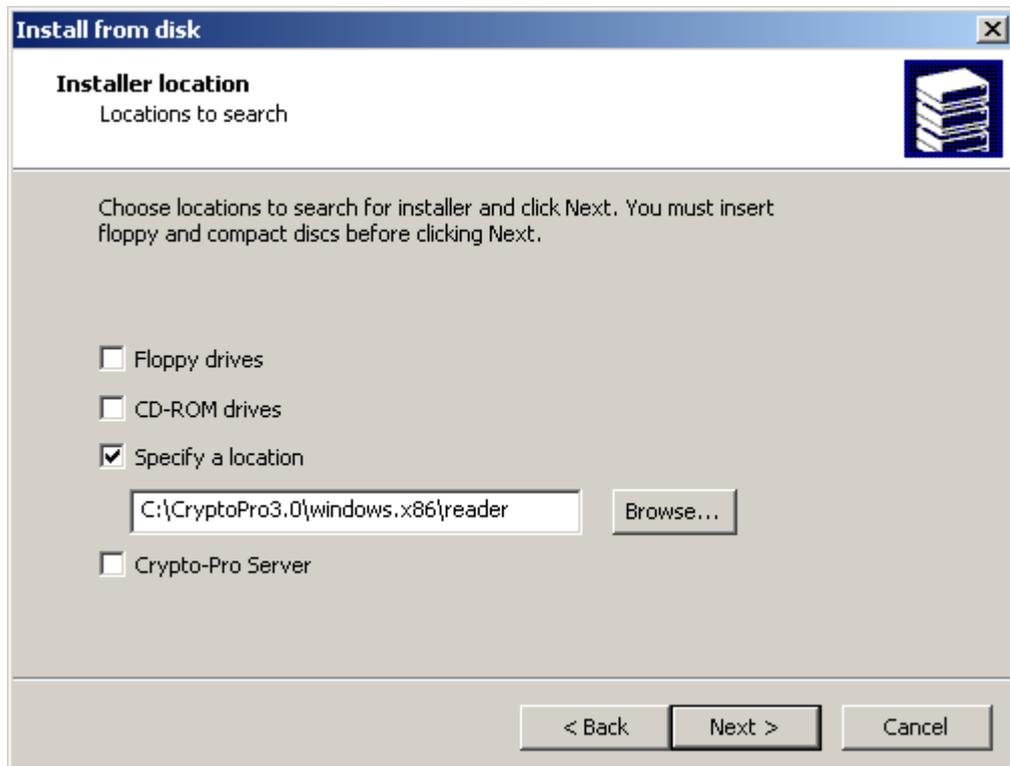


Рисунок 12

**Шаг 8:** из представленного списка выберите инсталлятор "считывателя Реестр" и нажмите кнопку Next:

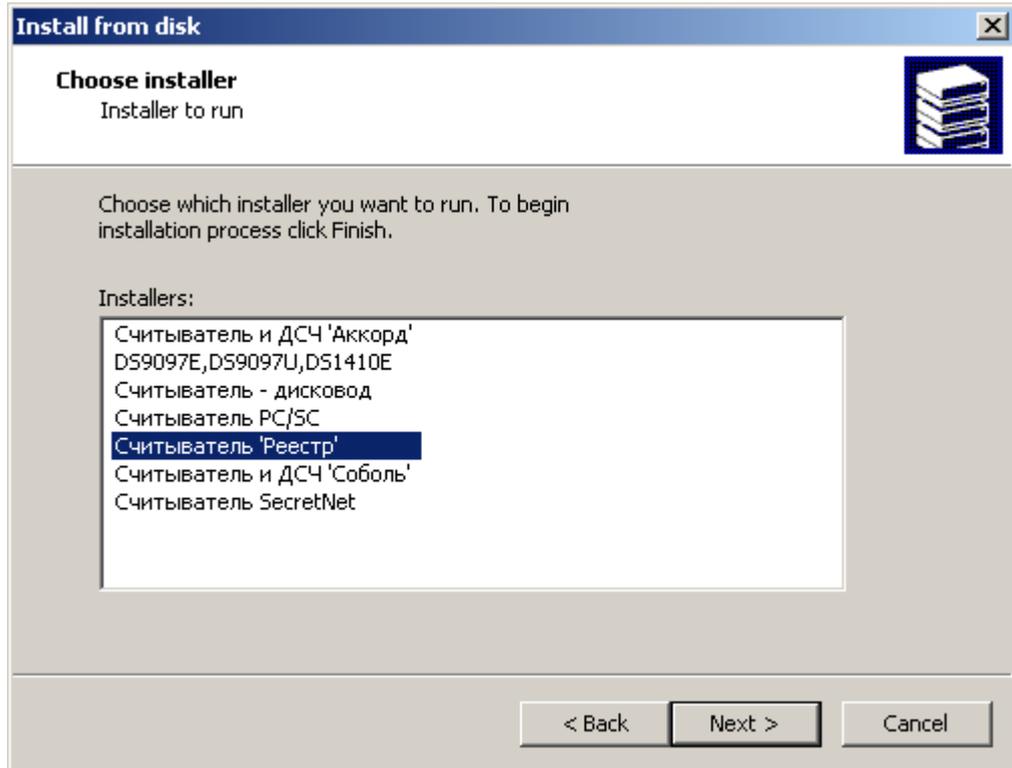


Рисунок 13

**Шаг 9:** по завершении выбора нажмите кнопку Finish:

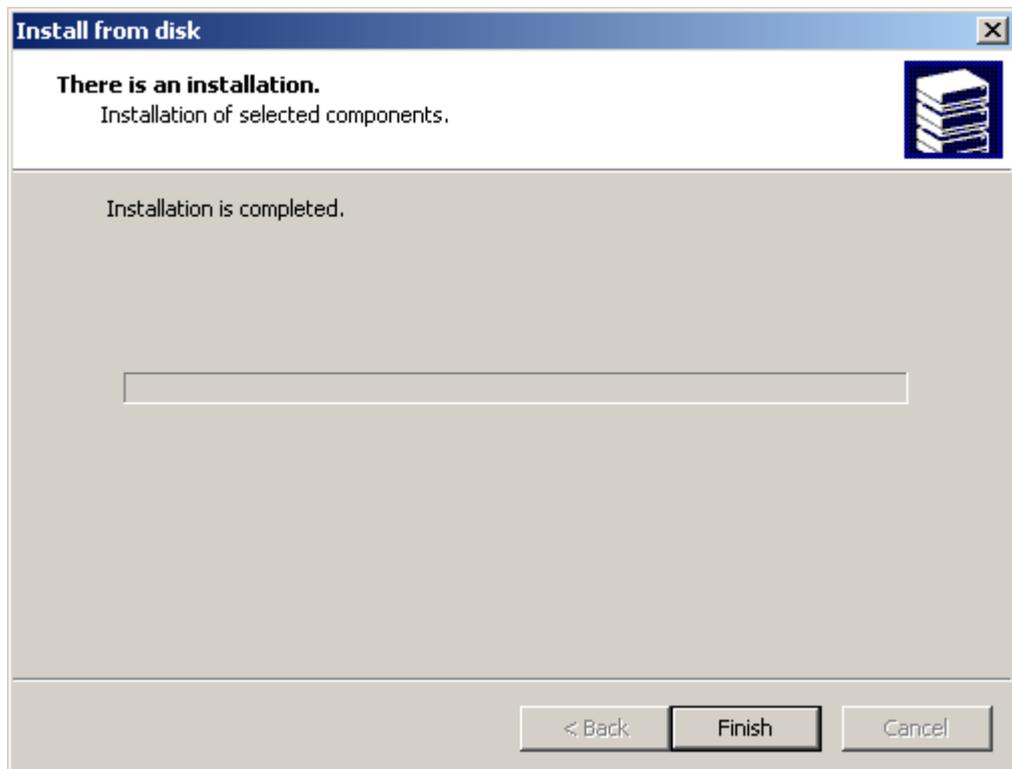


Рисунок 14

**Шаг10:** инсталлятор считывателя Registry добавлен в список имеющихся считывателей. Для запуска инсталляции нажмите кнопку Next:

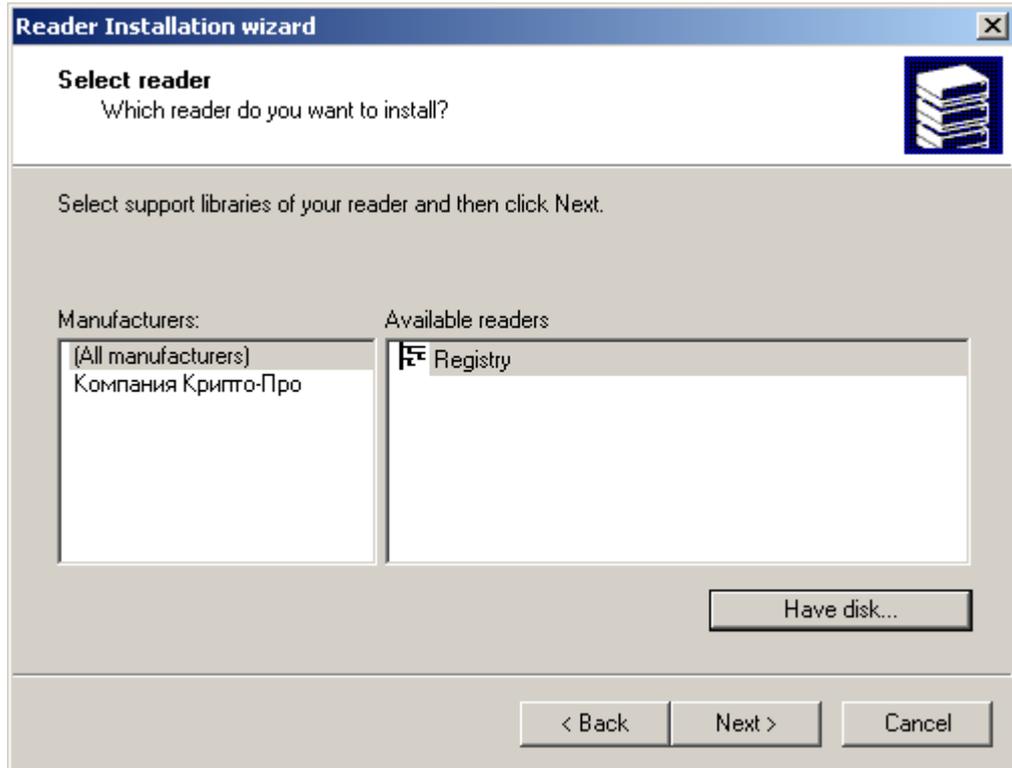


Рисунок 15

**Шаг11:** считывателю Registry можно присвоить имя и нажать кнопку Next:

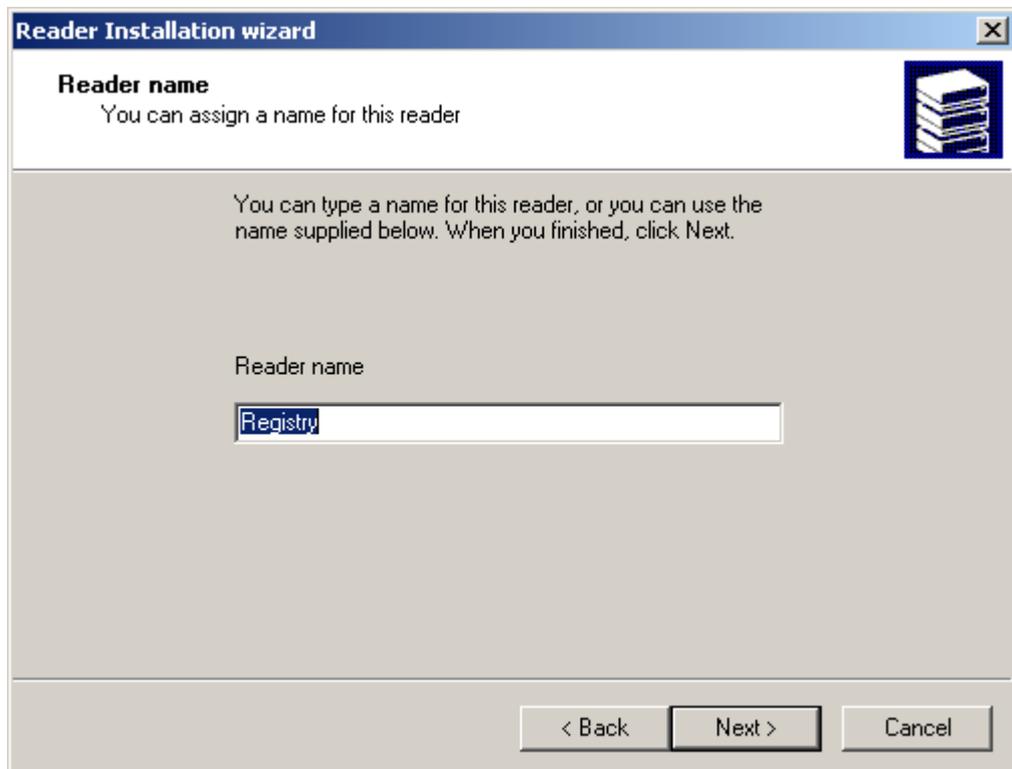


Рисунок 16

Шаг12: инсталляция считывателя Registry завершена, нажмите Finish:

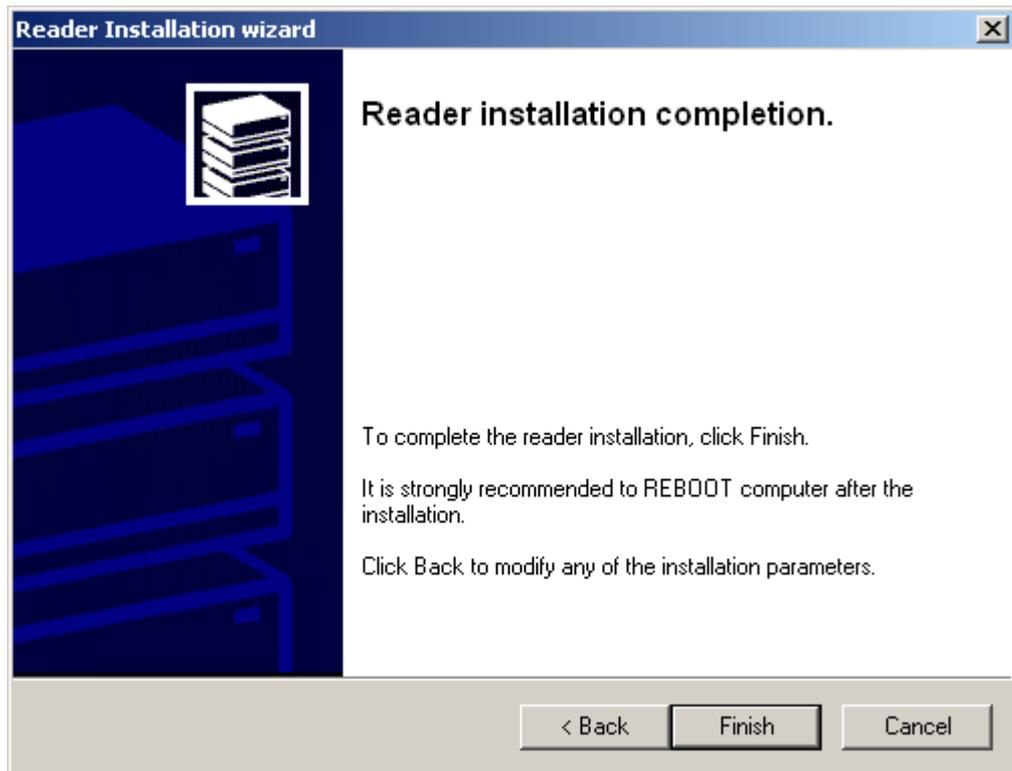


Рисунок 17

Шаг13: новый считыватель Registry добавлен в список инсталлированных, нажмите OK:



Рисунок 18

## 10.4.2. Установка считывателя и ключевого носителя eToken в "КриптоПро CSP 3.0"

Для установки внешнего ключевого носителя eToken нужно выполнить описанные ниже действия. Сначала устанавливаем ключевой считыватель для eToken.

**Шаг1:** запустить КриптоПро CSP: Start -Settings-Control Panel - CryptoPro CSP

**Шаг2:** в появившемся окне Properties войти во вкладку Hardware и нажать кнопку Configure readers (настроить считыватели):

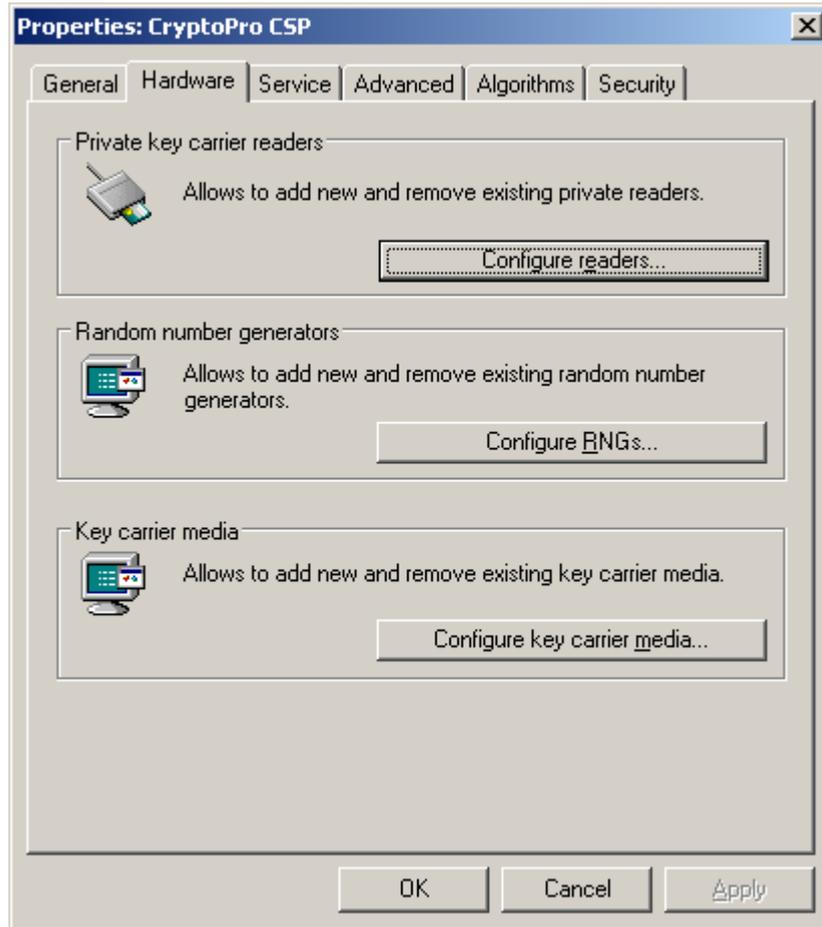


Рисунок 19

**Шаг3:** нажать кнопку Add, чтобы добавить новый ключевой считыватель:

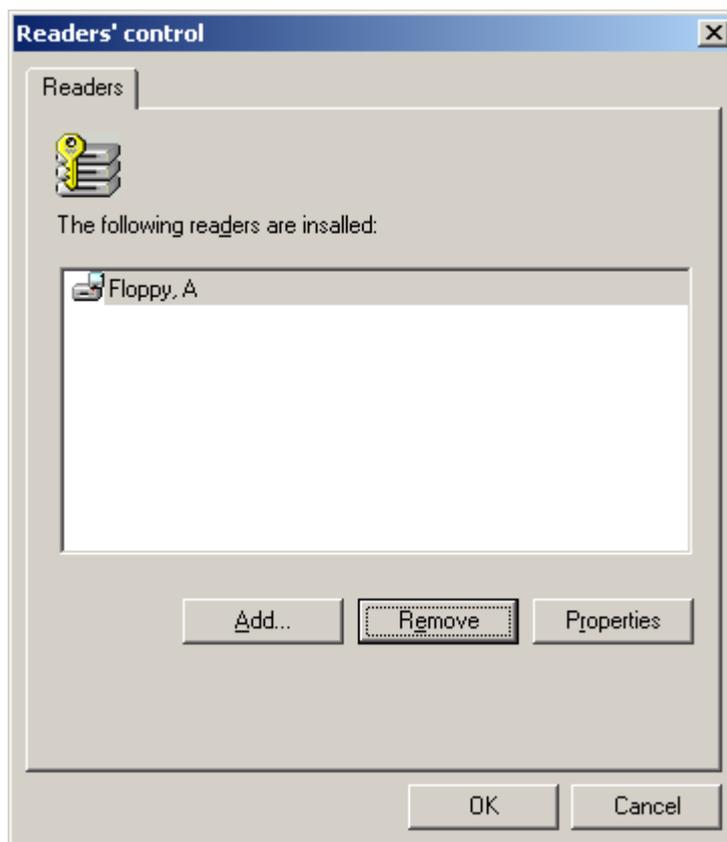


Рисунок 20

**Шаг4:** в окне визарда для инсталляции считывателя нажать кнопку Next :

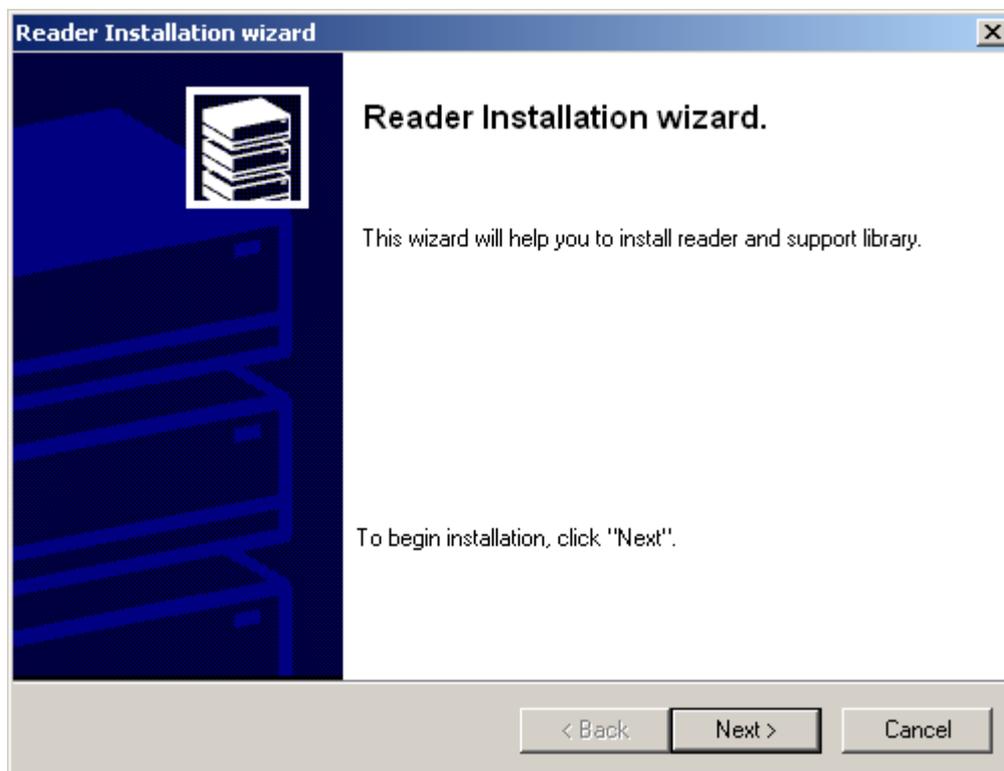


Рисунок 21

Шаг5 : для выбора библиотеки считывателя нажмите кнопку Have disk...

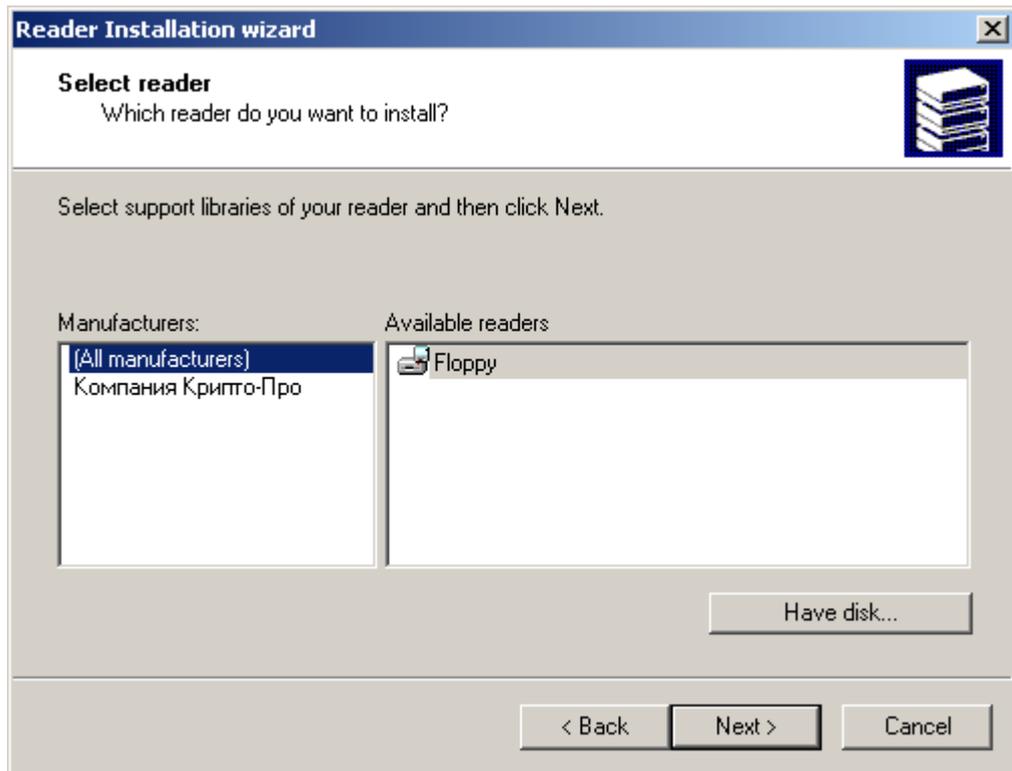


Рисунок 22

Шаг6 : и далее нажмите Next :

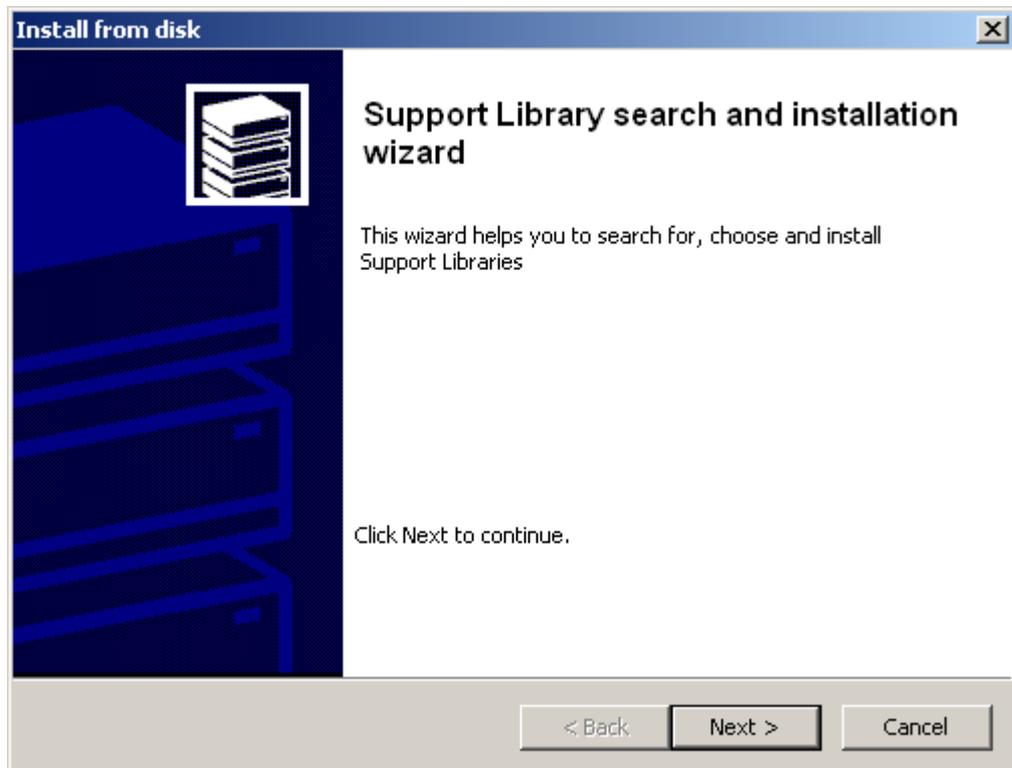


Рисунок 23

**Шаг 7:** укажите местоположение инсталлятора считывателя и нажмите Next :

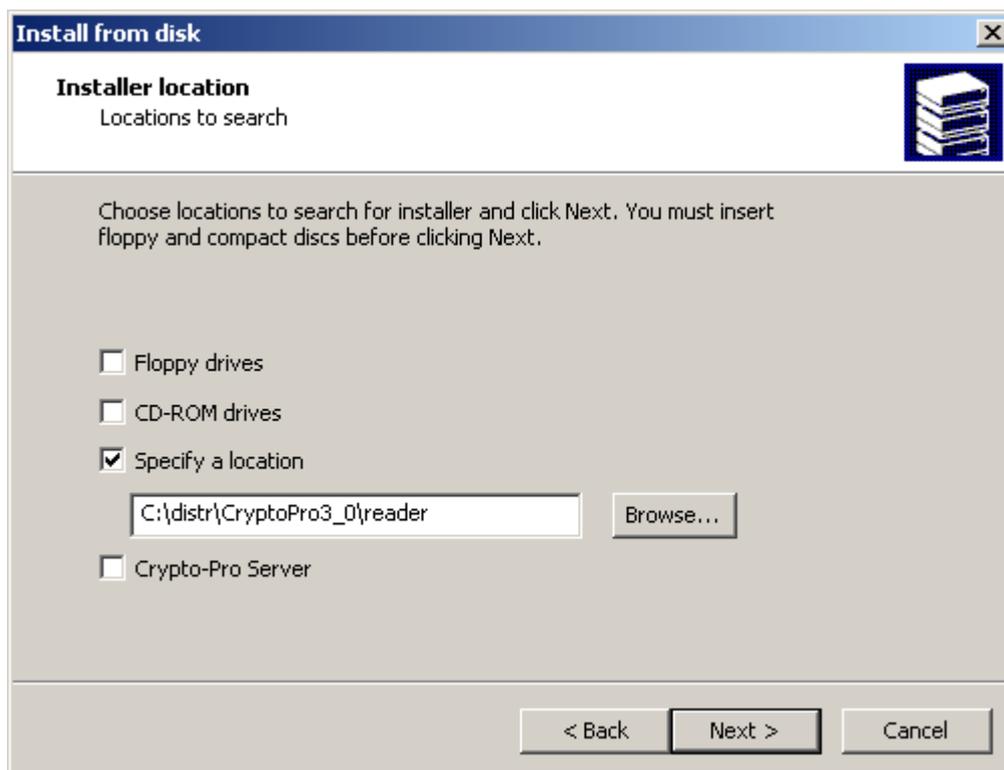


Рисунок 24

**Шаг 8:** из представленного списка выберите инсталлятор "Считыватель PC/SC" и нажмите кнопку Next :

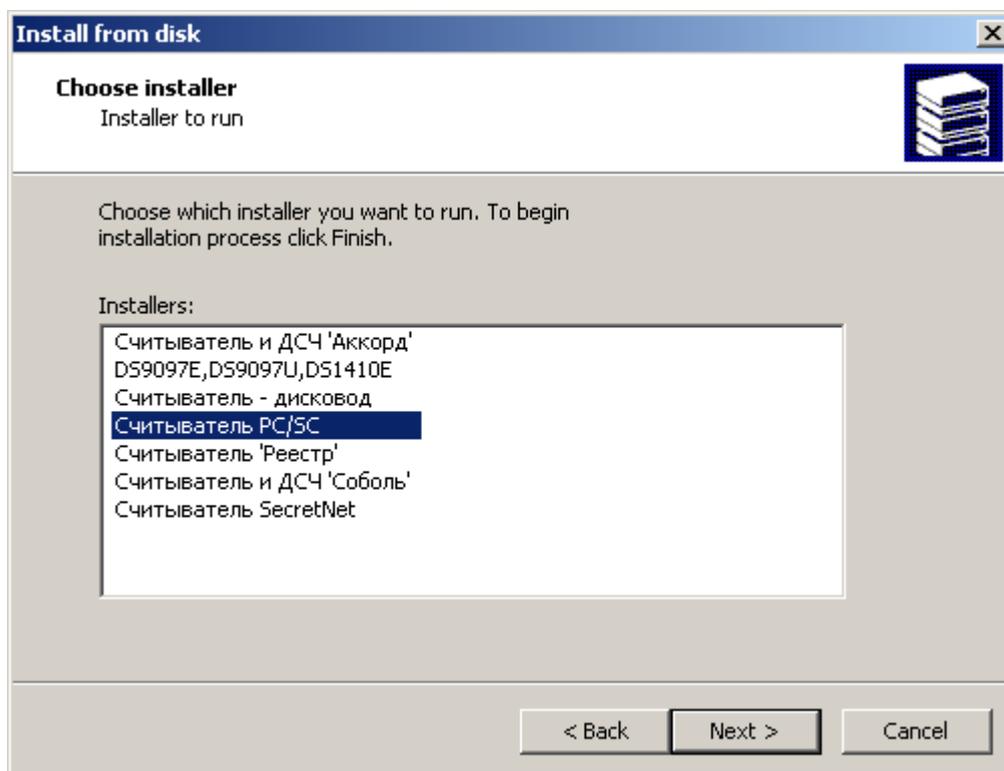


Рисунок 25

Шаг9: по завершению инсталляции нажмите кнопку Finish:

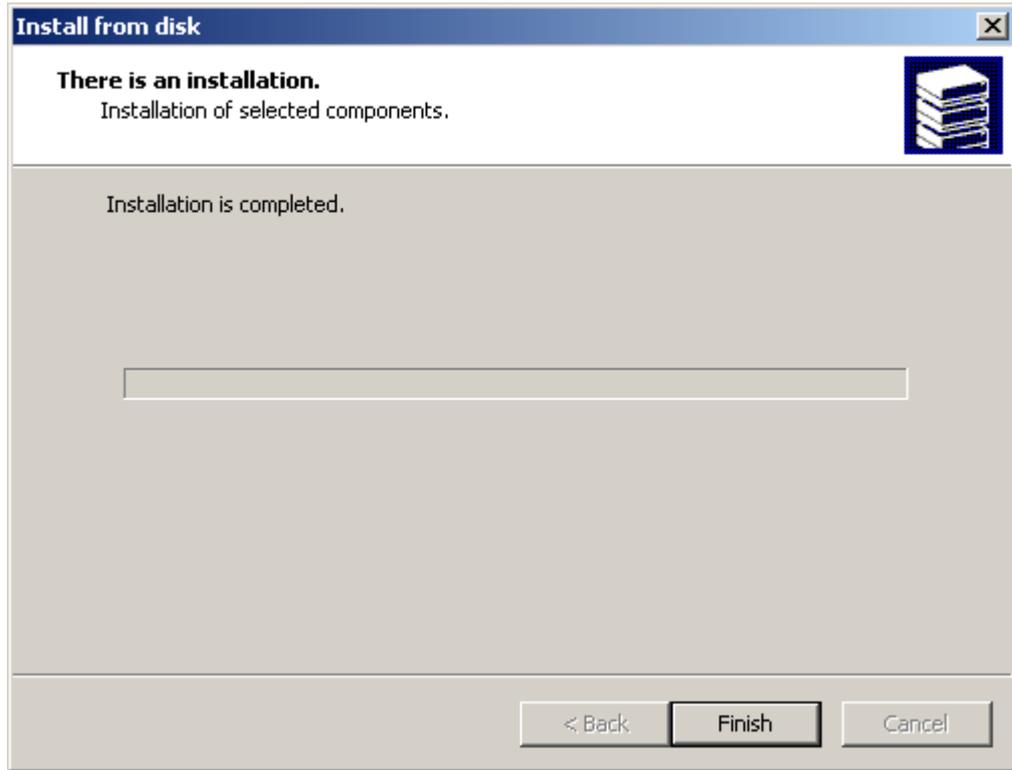


Рисунок 26

Шаг10: Два считывателя eToken добавлены в список имеющихся считывателей. Выделите один считыватель и нажмите кнопку Next:

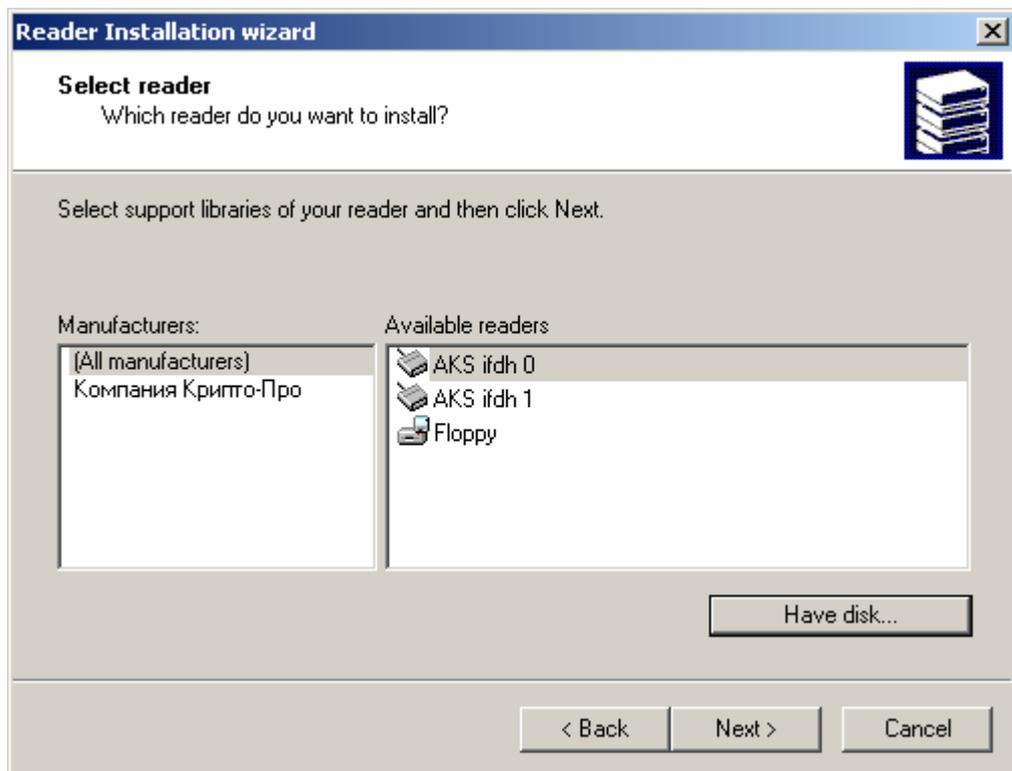


Рисунок 27

Шаг11: выделенному считывателю можно присвоить имя и нажать кнопку Next:

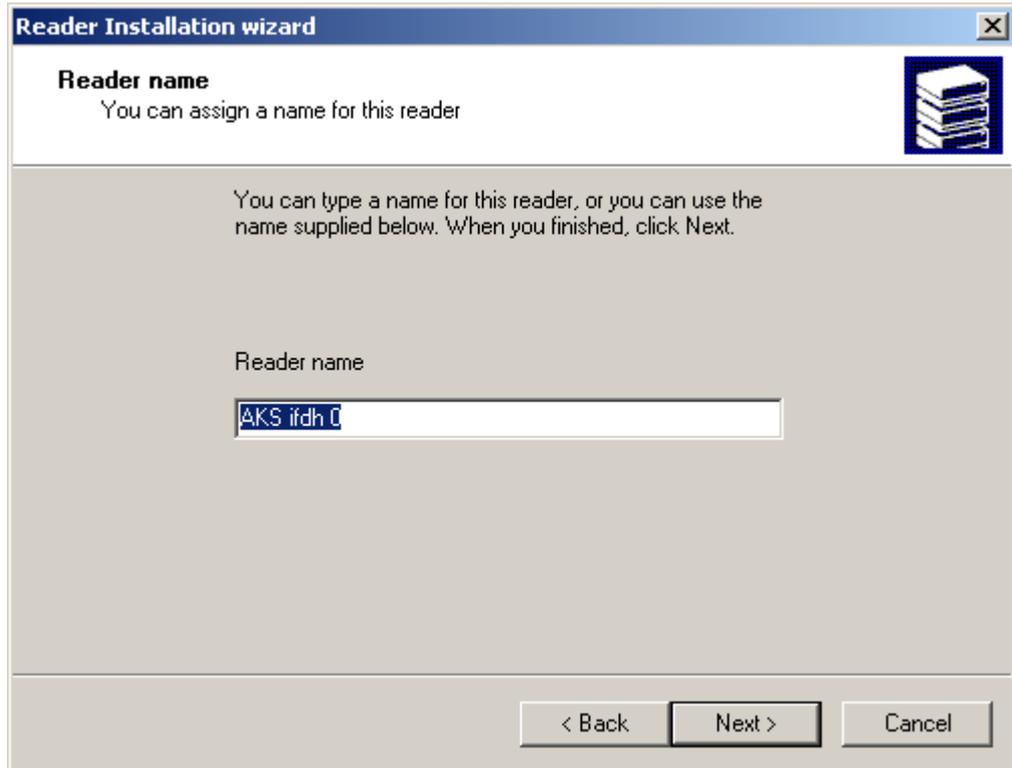


Рисунок 28

Шаг12: инсталляция выделенного считывателя завершена, нажмите Finish:

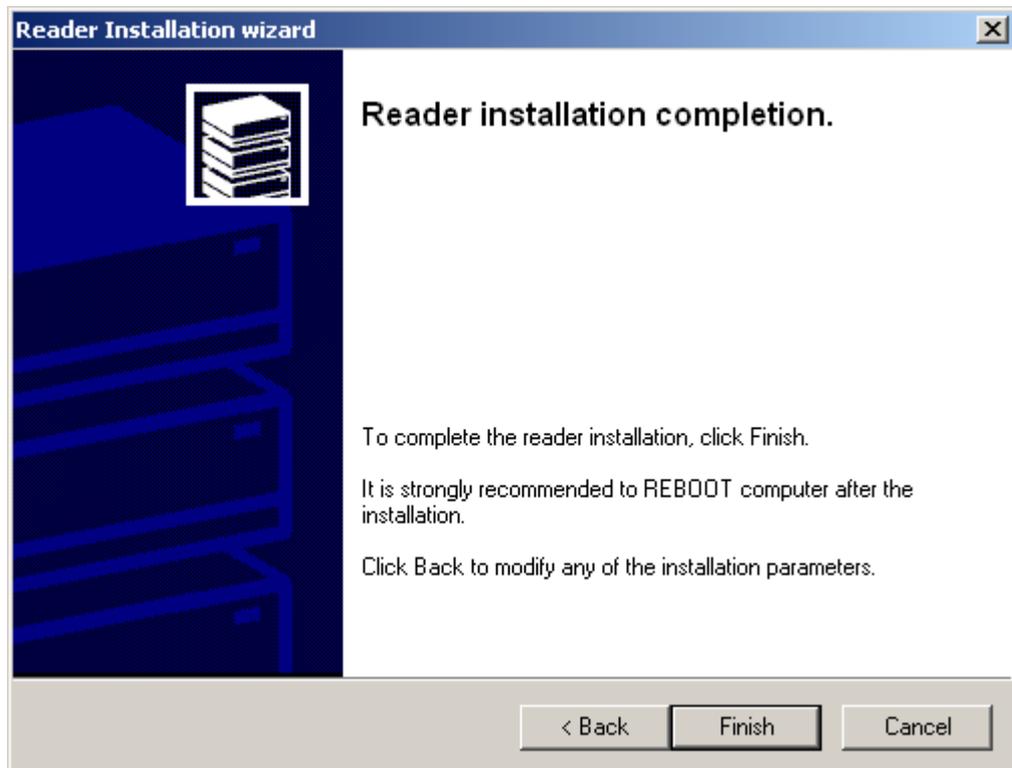


Рисунок 29

Шаг13: новый считыватель AKS ifdh 0 для eToken добавлен в список инсталлированных, нажмите OK:



Рисунок 30

Далее нужно установить ключевой носитель eToken.

**Шаг 14:** Во вкладке Hardware нажмите кнопку Configure key carrier media... (Рисунок 31) .

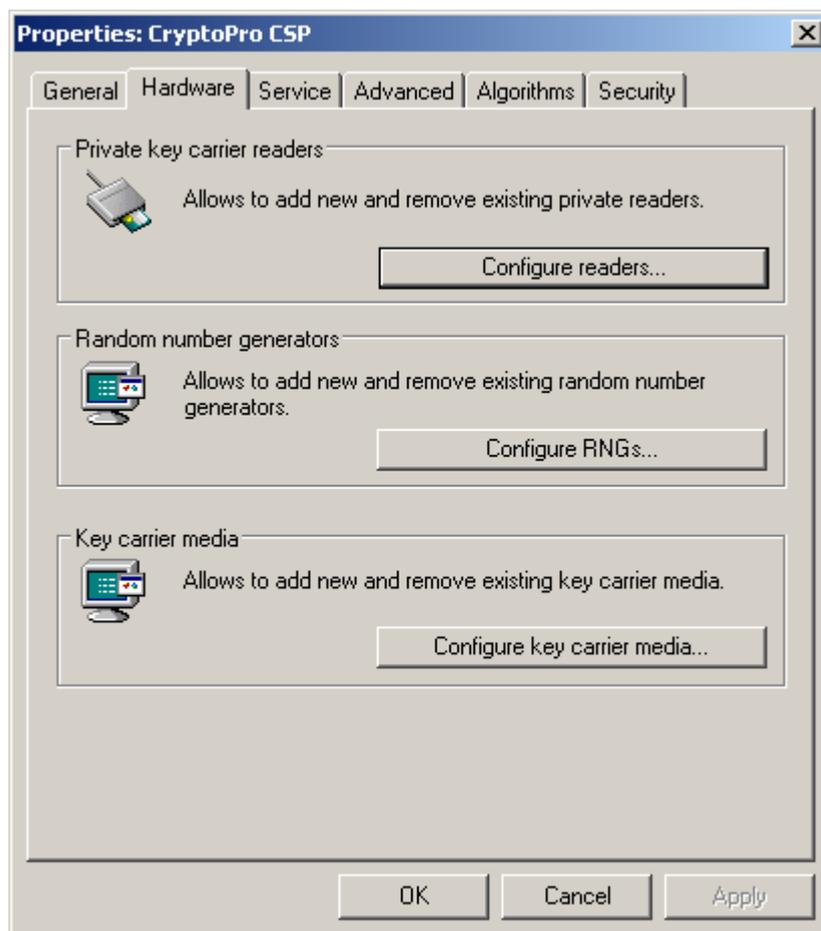


Рисунок 31

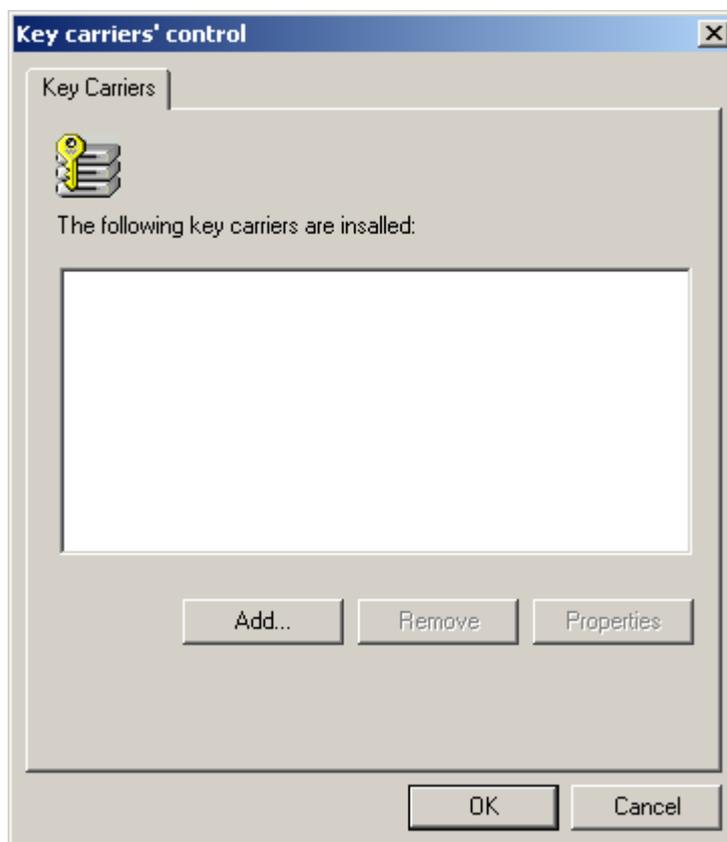


Рисунок 32

Шаг15: В следующем окне нажмите кнопку Add (Рисунок 32) .

Шаг16: Нажмите кнопку Next (Рисунок 33) :

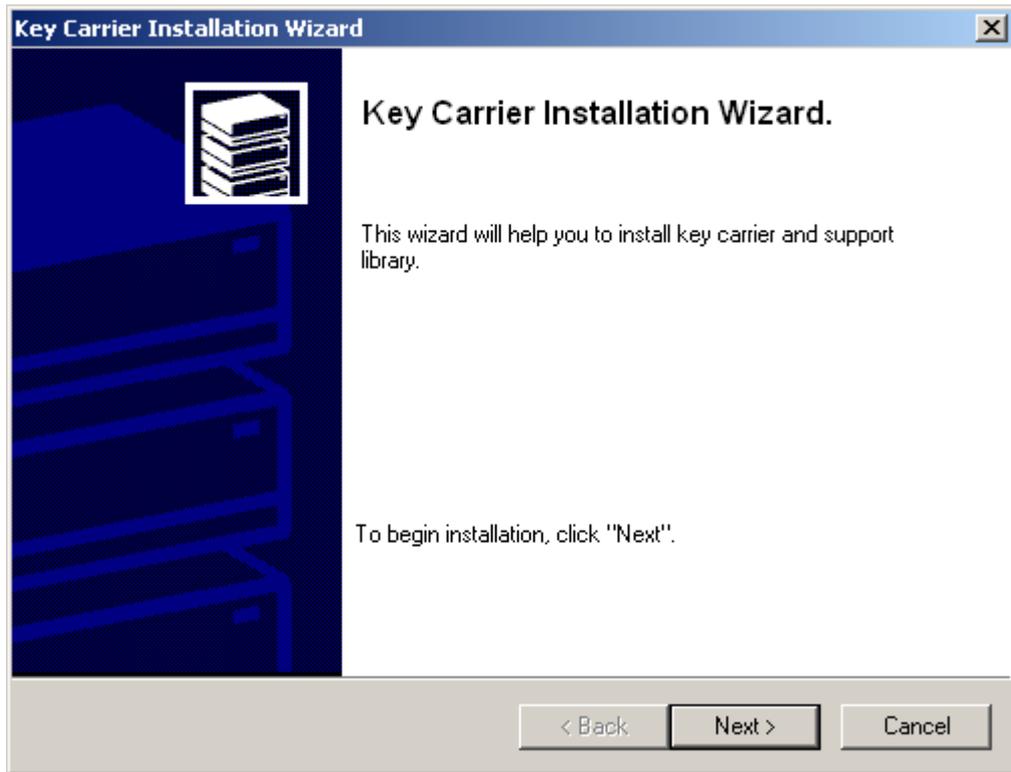


Рисунок 33

Шаг17: Для выбора библиотеки ключевых носителей нажмите кнопку Have disk...

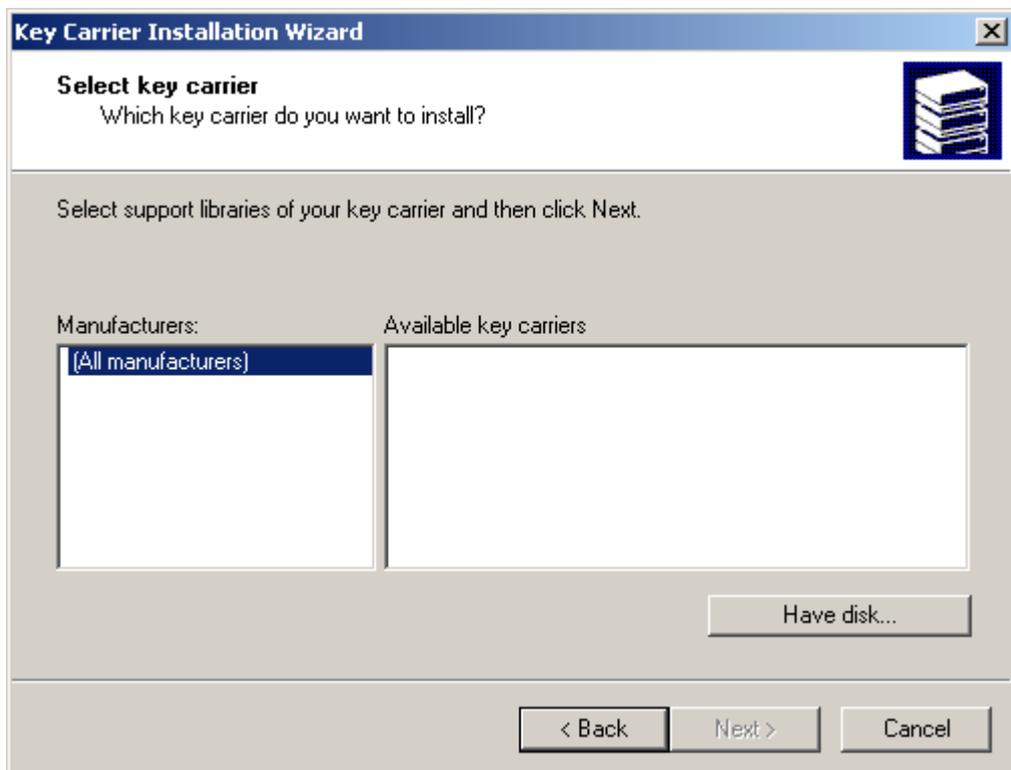


Рисунок 34

Шаг 18: Нажмите кнопку Next :

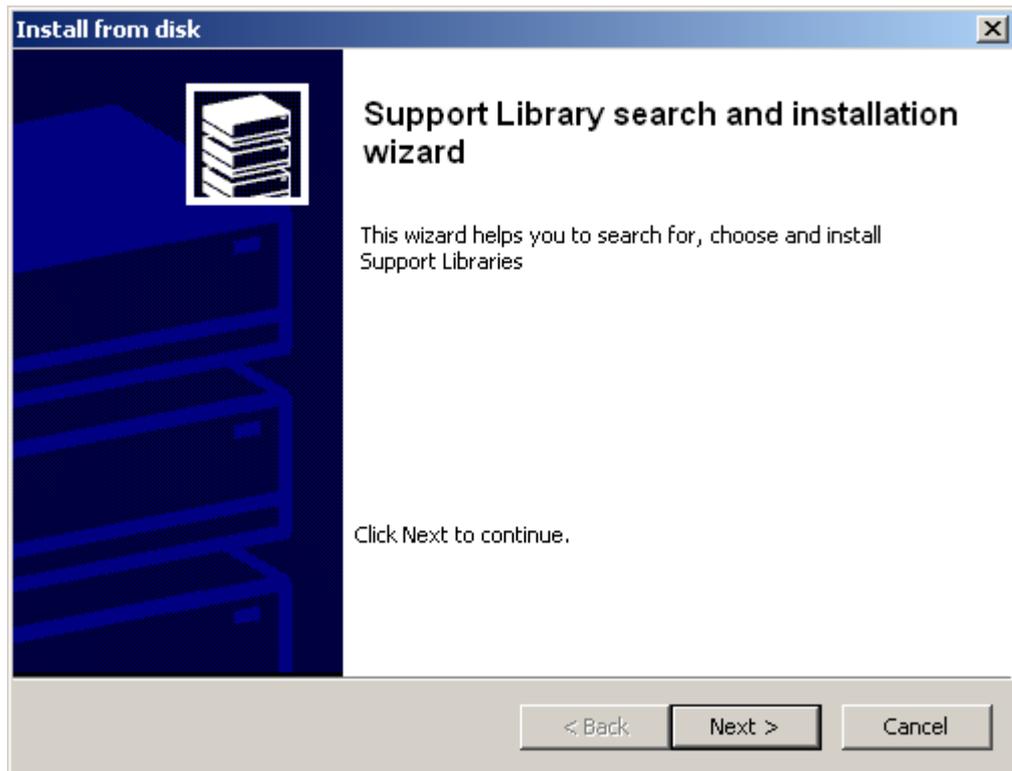


Рисунок 35

Шаг 19: Укажите размещение инсталлятора для ключевого носителя и нажмите кнопку Next :



Рисунок 36

**Шаг20:** Выделите ключевые носители – Смарт-карты eToken R2, PRO16, PRO32 и нажмите Next:

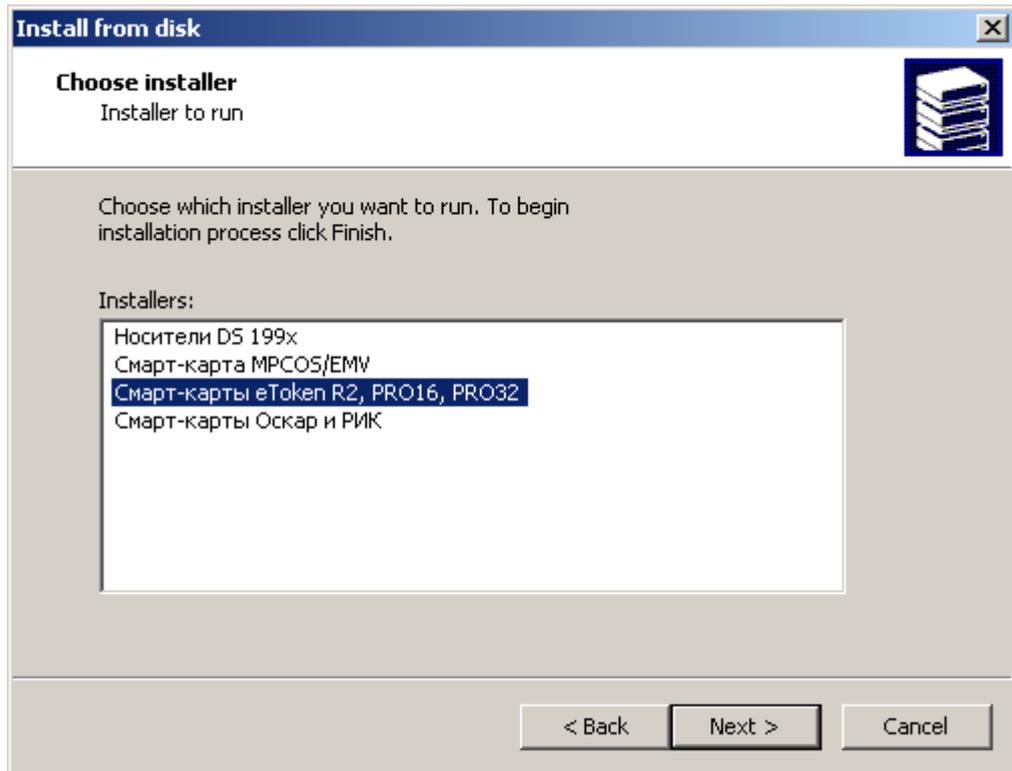


Рисунок 37

**Шаг21:** Нажмите кнопку Finish:

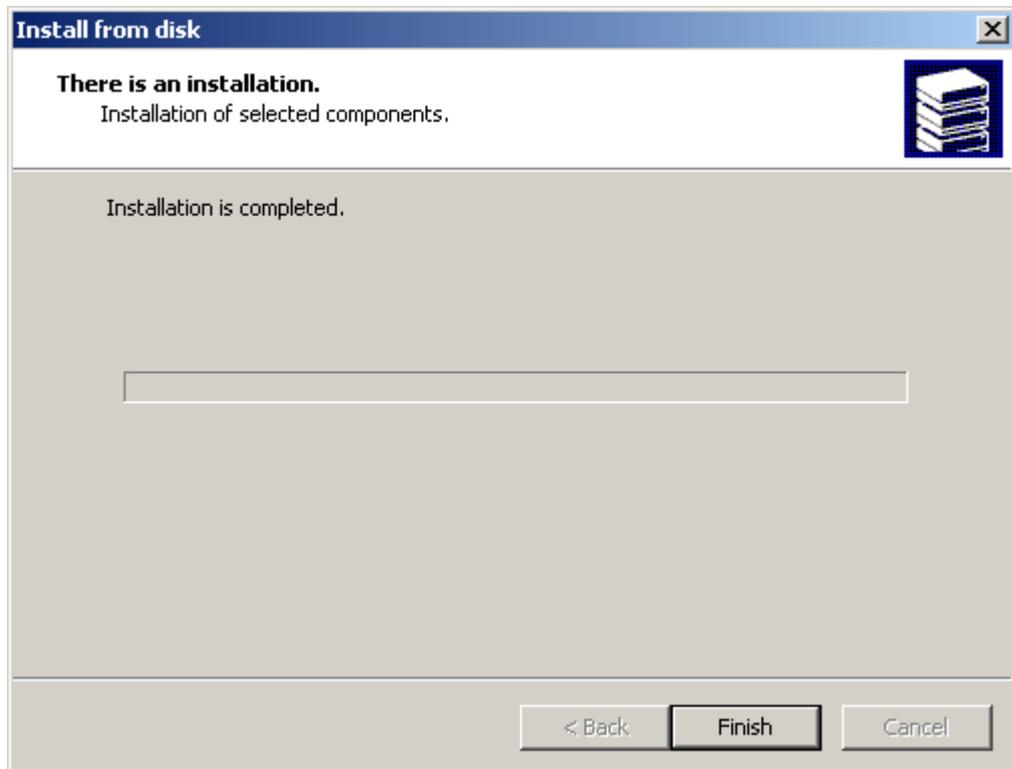


Рисунок 38

Шаг22 : Выберите ключевой носитель, например, eToken\_PRO16 и нажмите Next :

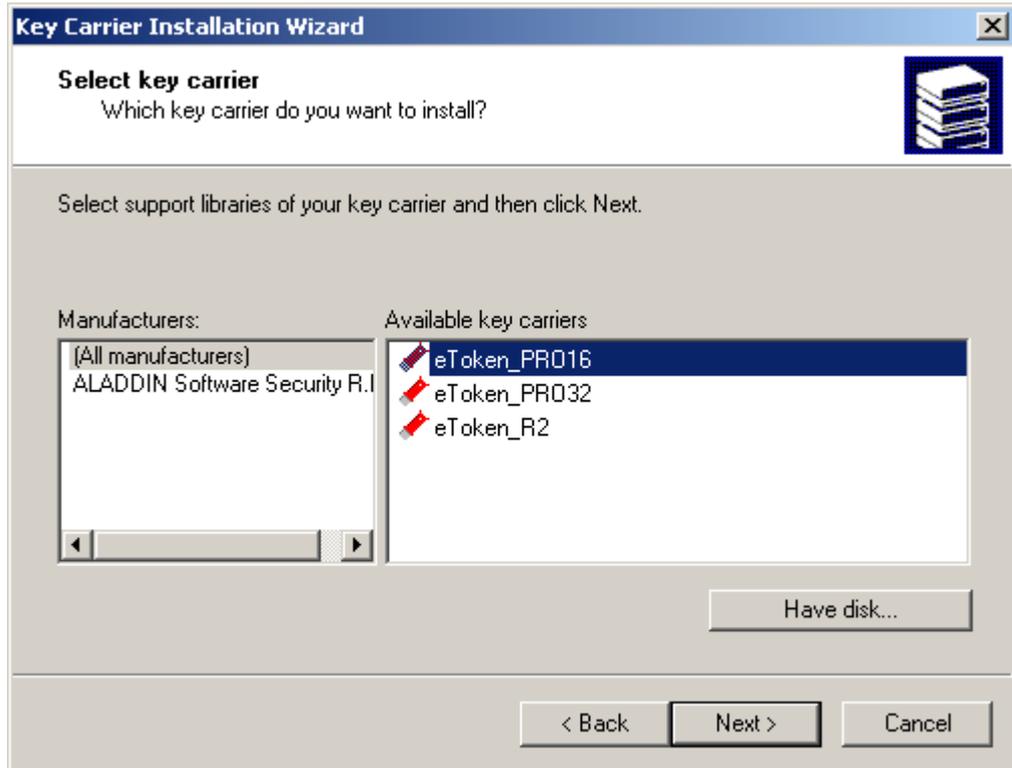


Рисунок 39

Шаг23 : Можно ввести имя этого носителя и нажать Next :

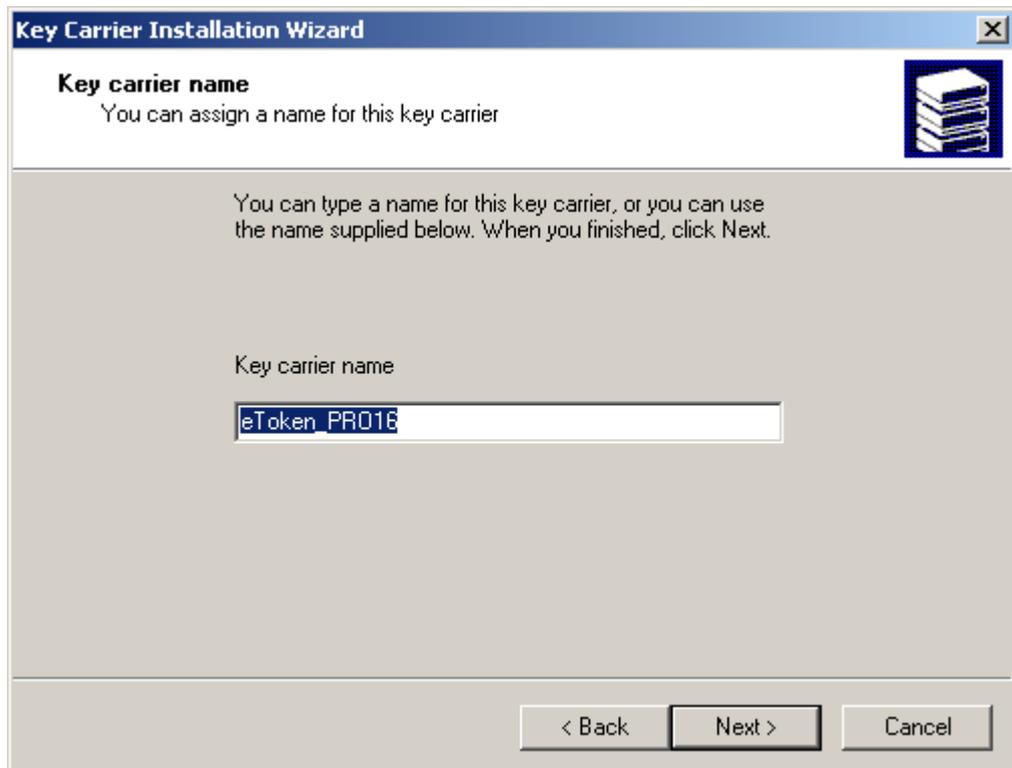


Рисунок 40

Шаг24 : Нажмите Next :

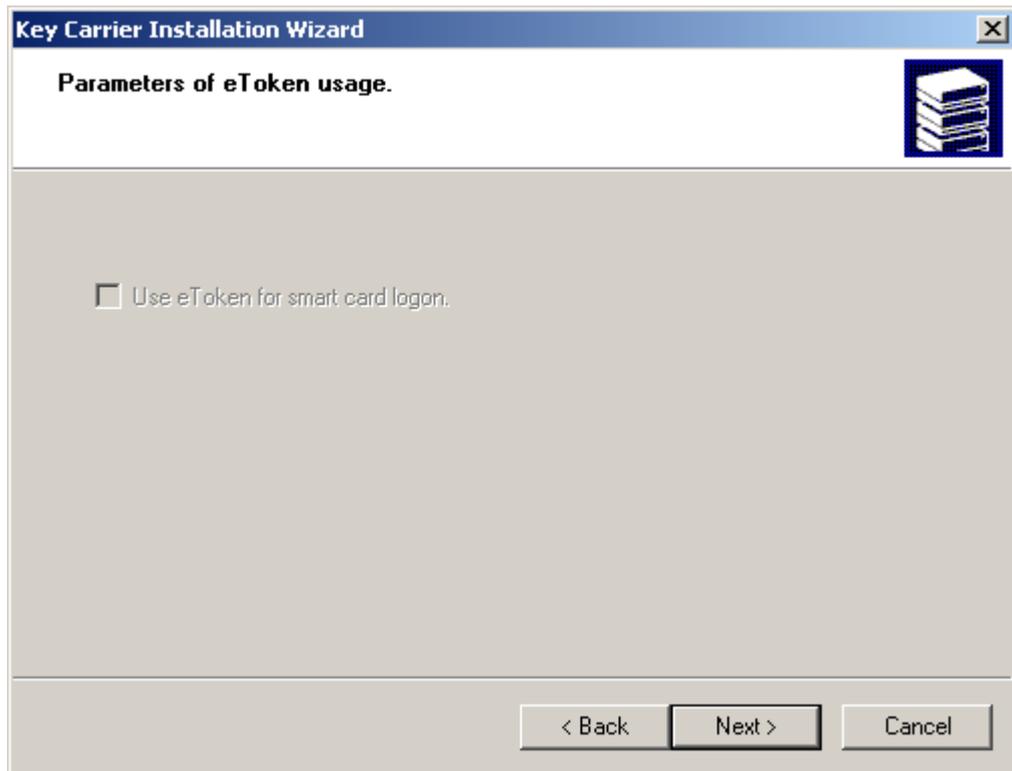


Рисунок 41

Шаг25 : Инсталляция ключевого носителя закончена, нажмите Finish.

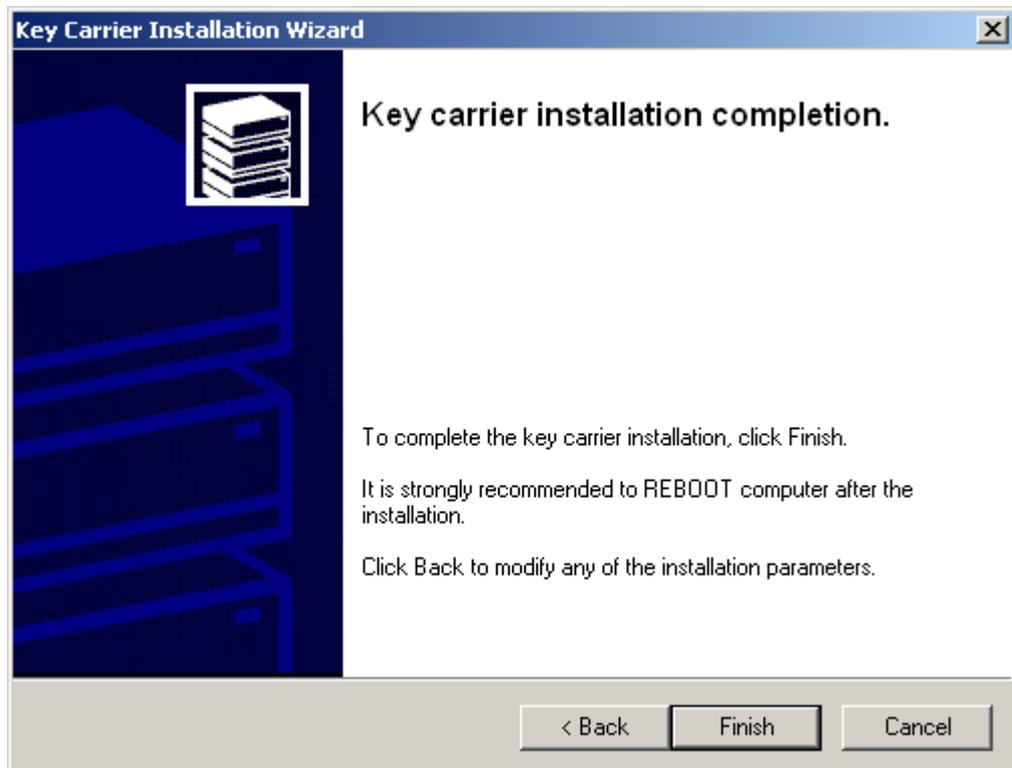


Рисунок 42

Таким образом, инсталлированы ключевой считыватель и ключевой носитель информации для eToken.

### 10.4.3. Установка и настройка Удостоверяющего Центра. Создание СА сертификата

Перед созданием ключевой пары и создания запроса на локальный сертификат опишем как создать Удостоверяющий Центр (Центр Сертификации – СА) 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 2000 (2003) Server (SP4) и СКЗИ «КриптоПро CSP 3.0». Сервис Internet Information Services (IIS) должен быть включен.

Также установите ключевой носитель, например Registry, для хранения контейнера с секретным ключом СА сертификата.

Для инсталляции Удостоверяющего Центра Microsoft Certification Authority выполните следующие шаги:

**Шаг 1:** установите сертификатный сервис: в окне установки компонент Windows (Start-Settings-Control Panel-Add/Remove Programs-Add/Remove Windows Components) установите флажок напротив Certificate Services и нажмите Next:

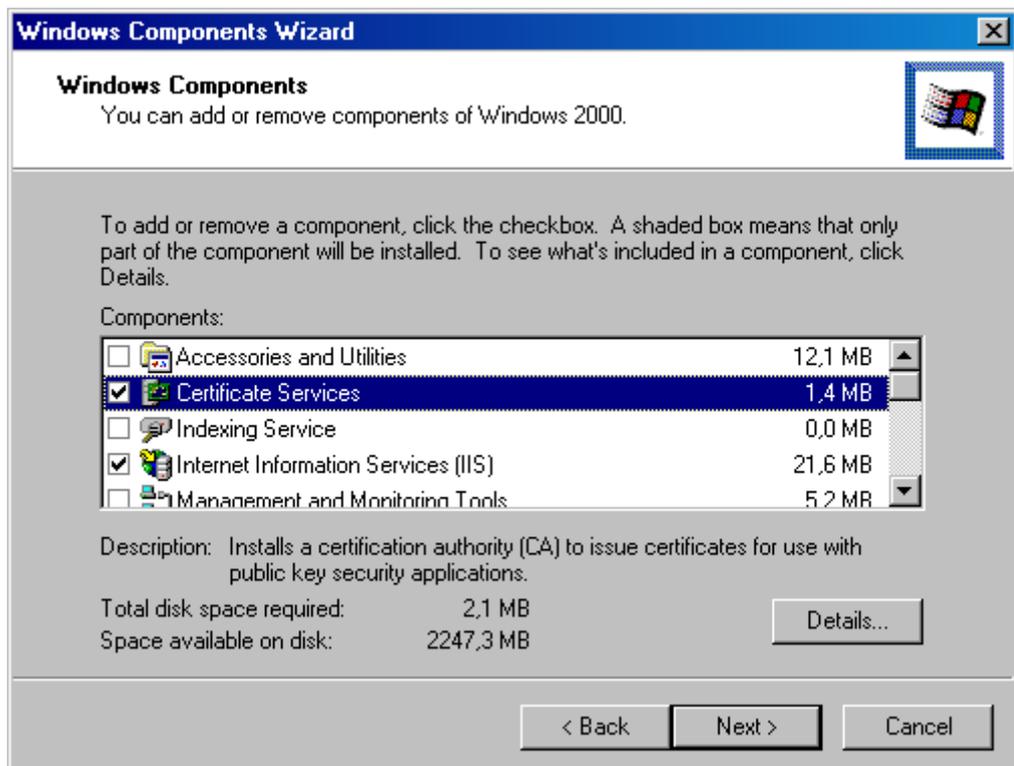


Рисунок 43

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

**Шаг2:** в следующем окне выберите Удостоверяющий Центр с корневым CA сертификатом: поставьте переключатель в положение Stand-alone root CA. Установите флажок Advanced options и нажмите Next:

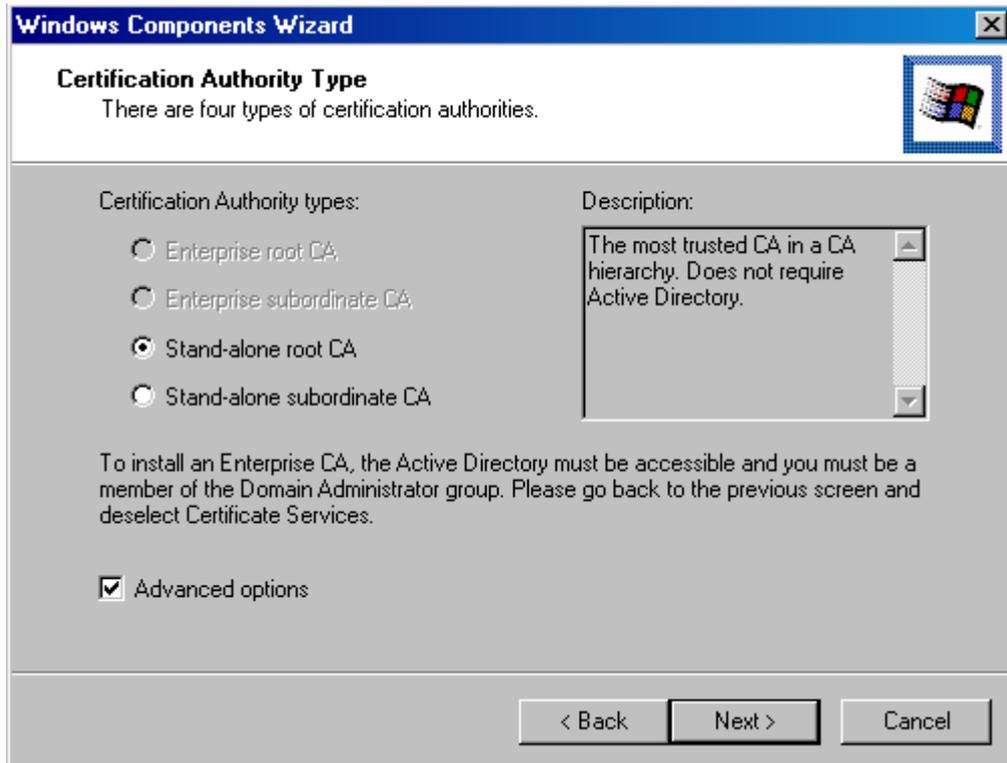


Рисунок 44

**Шаг3:** в качестве криптопровайдера выберите Crypto-Pro GOST R34.10-2001 Cryptographic Service, а в качестве хэш-алгоритма - GOST R34.11-94 и нажмите Next:

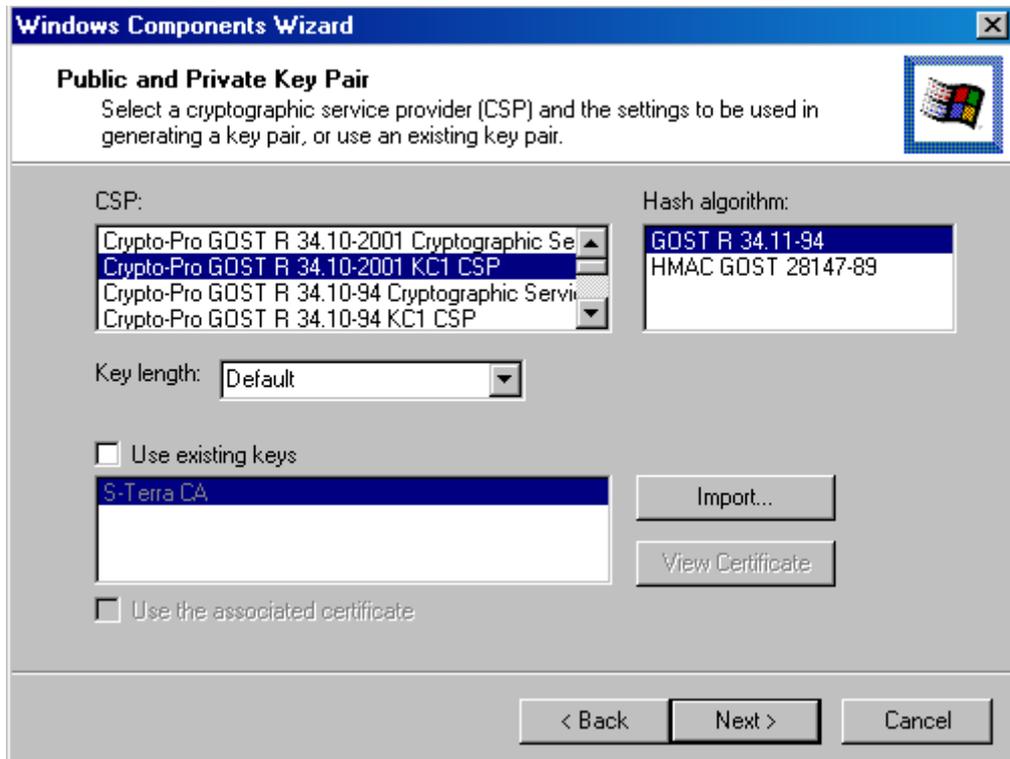


Рисунок 45

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

**Windows Components Wizard**

**CA Identifying Information**  
Enter information to identify this CA

CA name: S-Terra Center CA

Organization: S-Terra

Organizational unit: Center Devel

City: Moscow

State or province: Country/region: RU

E-mail: ca@s-terra.com

CA description: Test certificate

Valid for: 2 Years Expires: 9/27/2009 3:19 PM

< Back Next > Cancel

Рисунок 46

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

**Windows Components Wizard**

**CryptoPro CSP**

Insert empty carrier media  
S-Terra Center CA.

Details

Readers: Registry

Carrier media inserted:

Status:

OK Cancel Details <<

Рисунок 47

**Шаг 6:** подвигайте мышкой или нажмите любую клавишу пока происходит создание ключевой пары для CA сертификата:

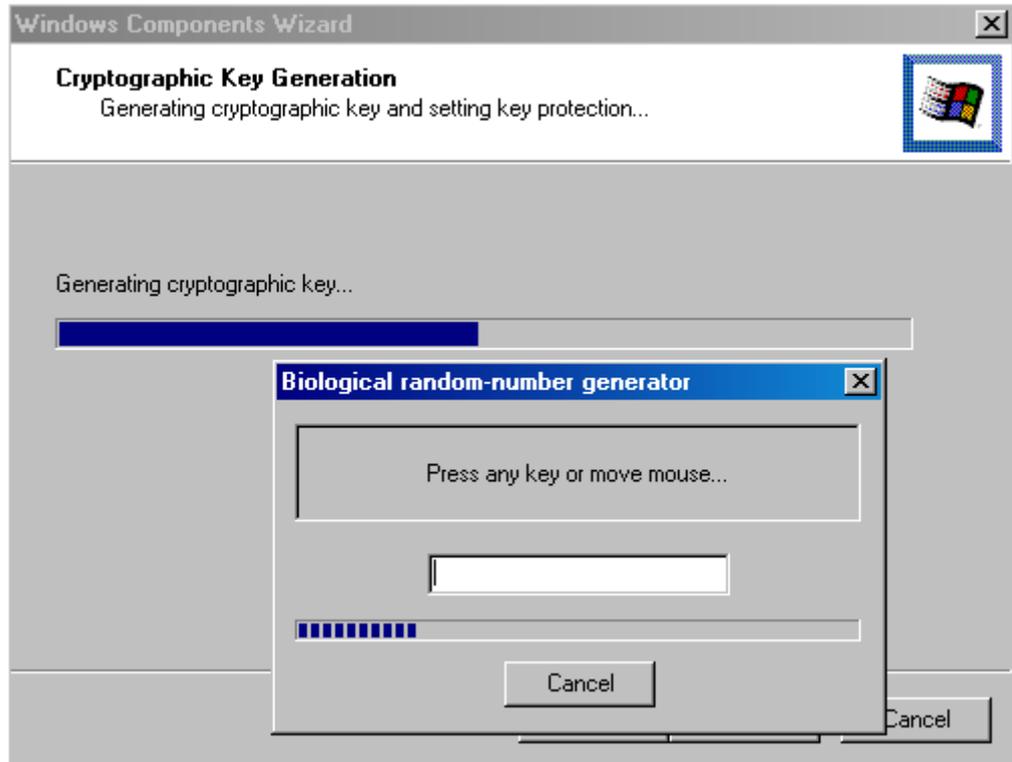


Рисунок 48

**Шаг 7:** задайте пароль к ключевому контейнеру и нажмите ОК:

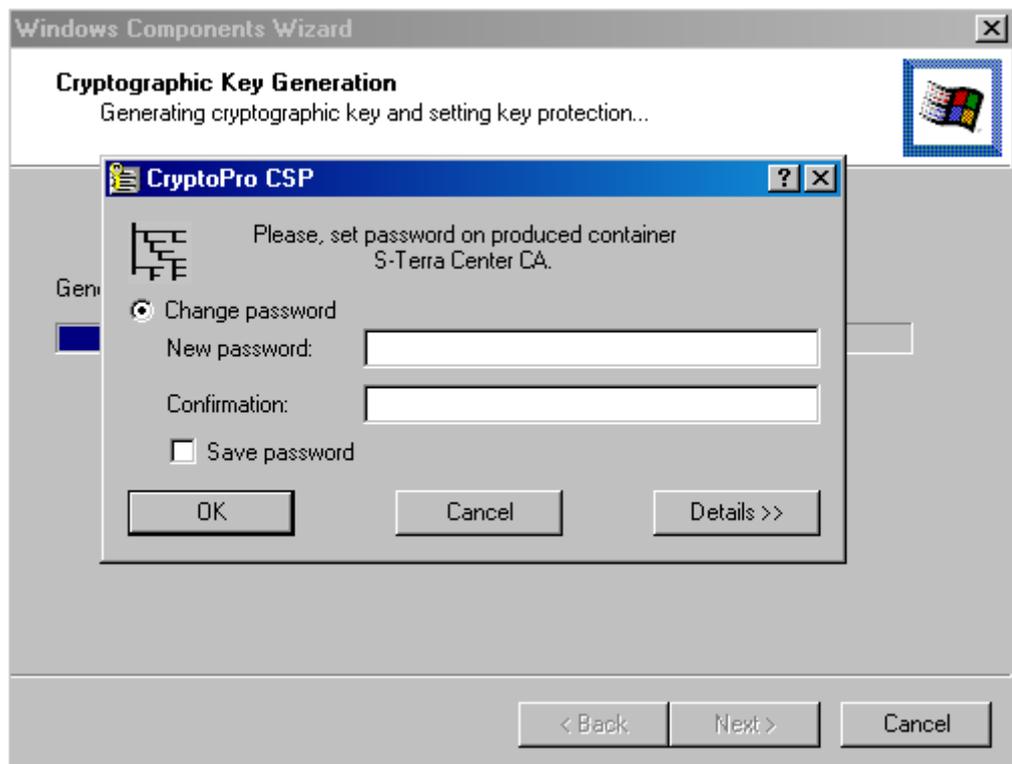


Рисунок 49

**Шаг 8:** в следующем окне введите еще раз пароль для контейнера с секретным ключом для CA сертификата и нажмите ОК.

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

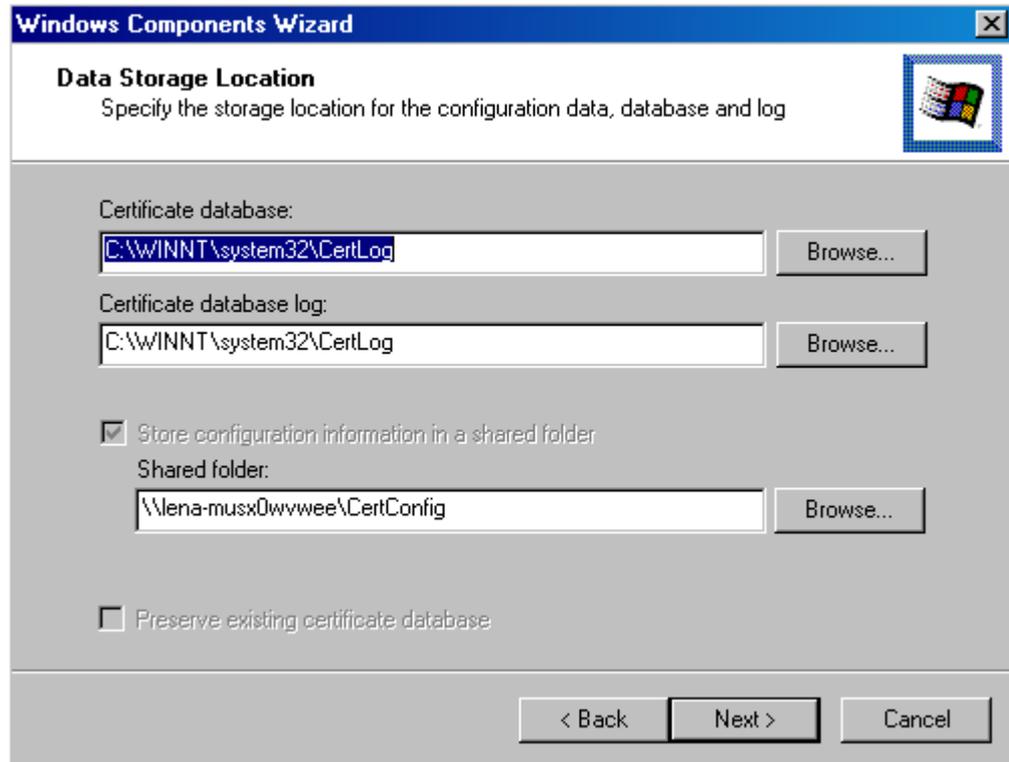


Рисунок 50

**Шаг 10:** для остановки Internet Information Services **НАЖМИТЕ** OK:

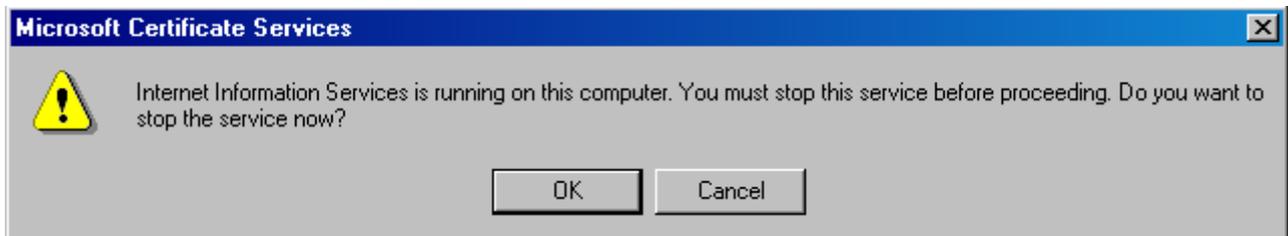


Рисунок 51

Шаг11: введите еще раз пароль к контейнеру с секретным ключом CA сертификата и нажмите ОК:

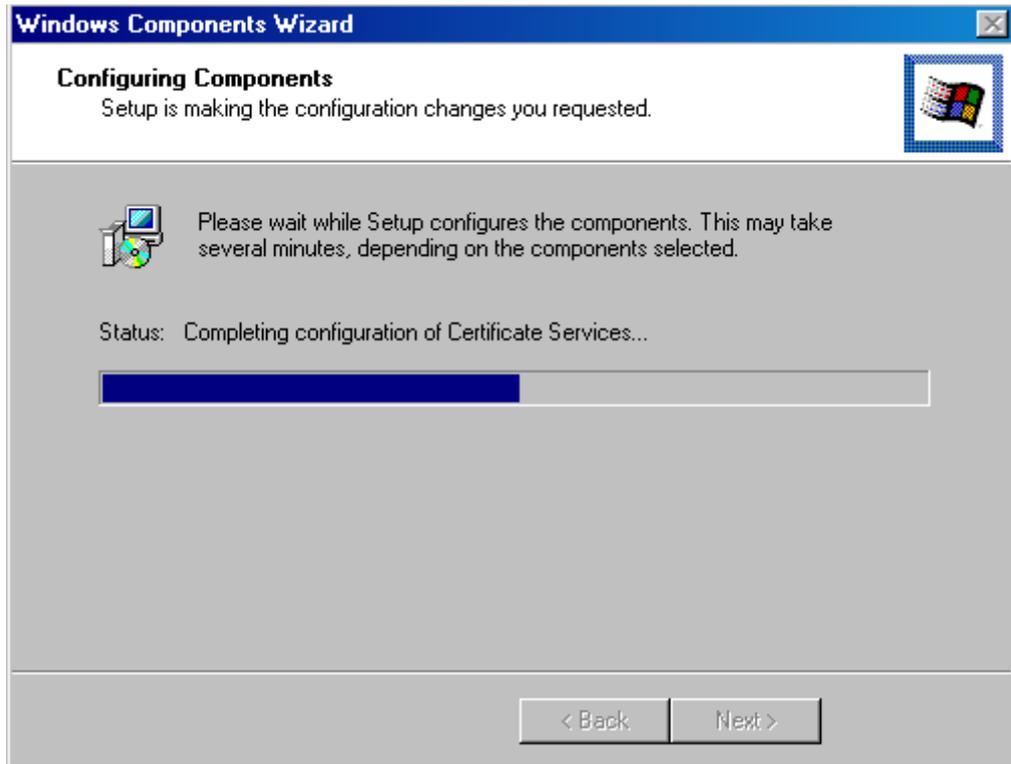


Рисунок 52

Шаг12: инсталляция Удостоверяющего Центра завершена, нажмите Finish.



Рисунок 53

**Шаг 13:** для автоматического создания подписываемых сертификатов войдите сначала в Certificate Authority (Start-Settings-Control Panel-Administrative Tools-Certificate Authority), выделите центр CA и нажмите Properties:

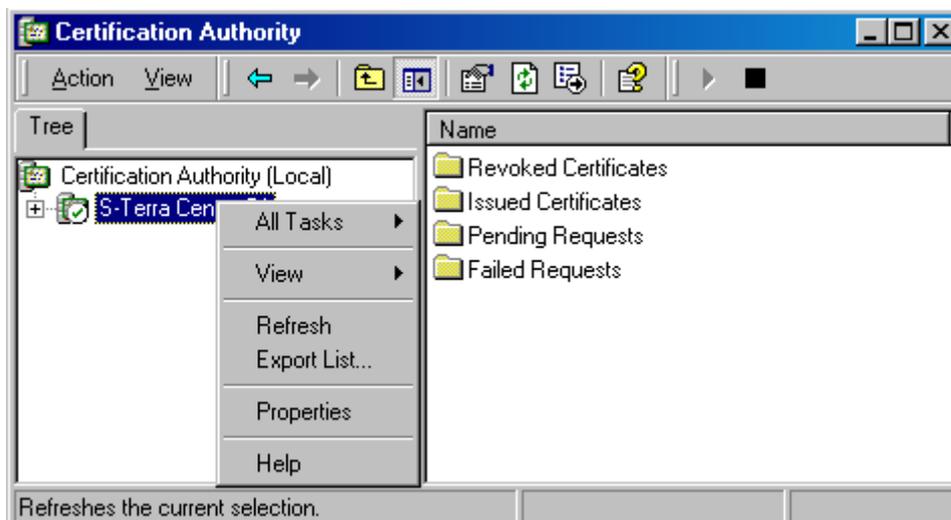


Рисунок 54

**Шаг 14:** 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:

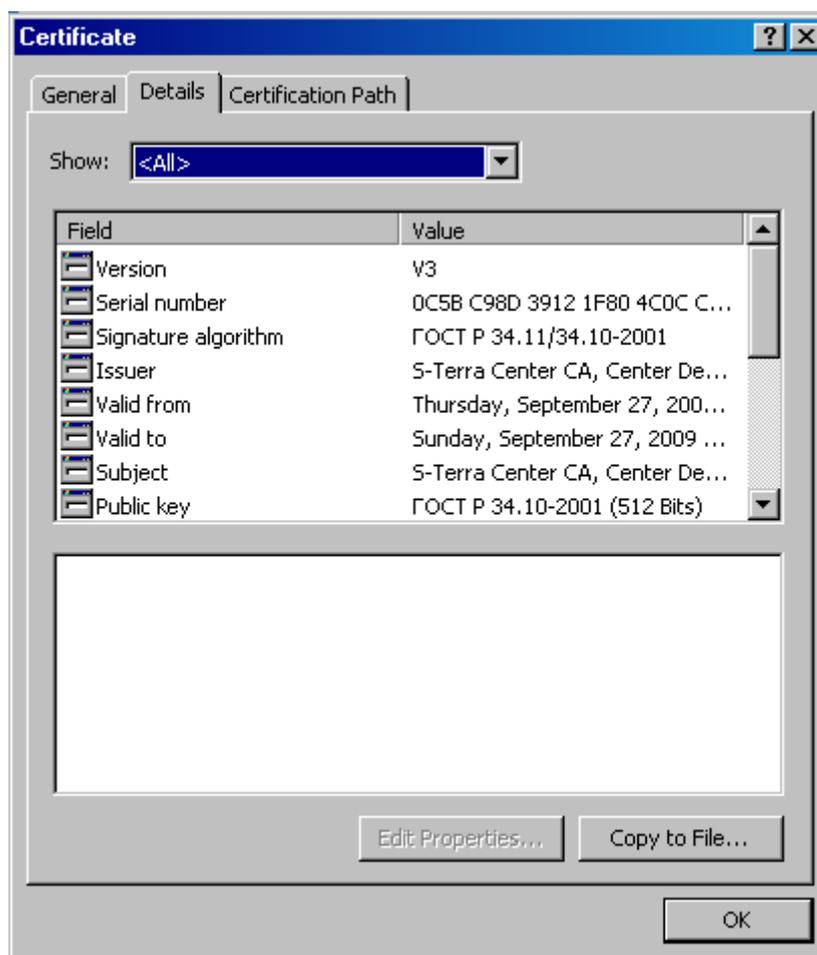


Рисунок 55

**Шаг15:** выберите формат для сертификата – установите переключатель в первое положение:

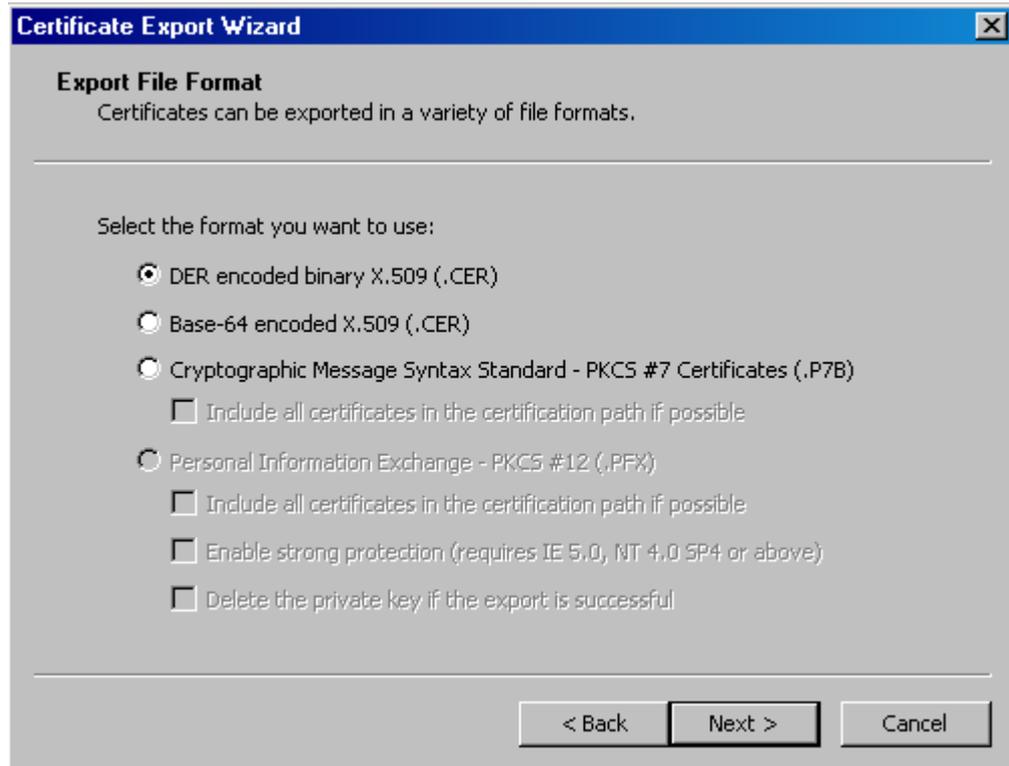


Рисунок 56

**Шаг16:** экспортируйте CA сертификат на дискету, указав имя файла:

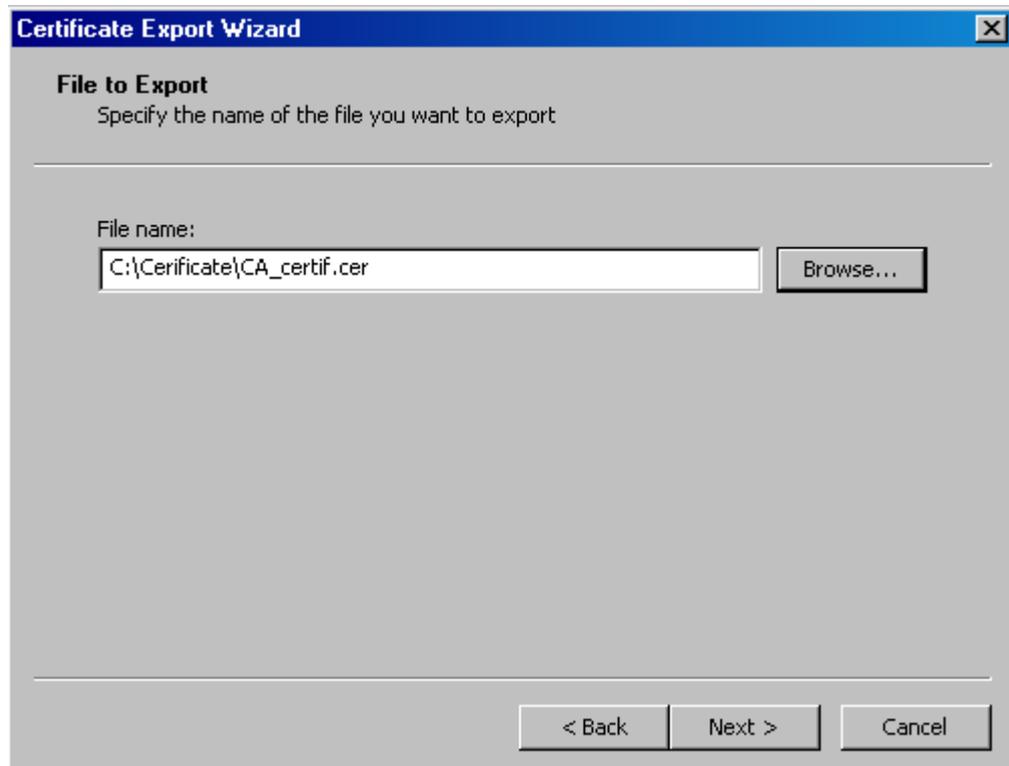


Рисунок 57

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



Рисунок 58

Шаг 18: для автоматического создания подписываемых сертификатов на сервере проведите некоторые настройки: войдите в сертификационный центр (Start – Settings – Control Panel – Administrative Tools – Certification Authority), выделите центр CA и нажмите Properties:

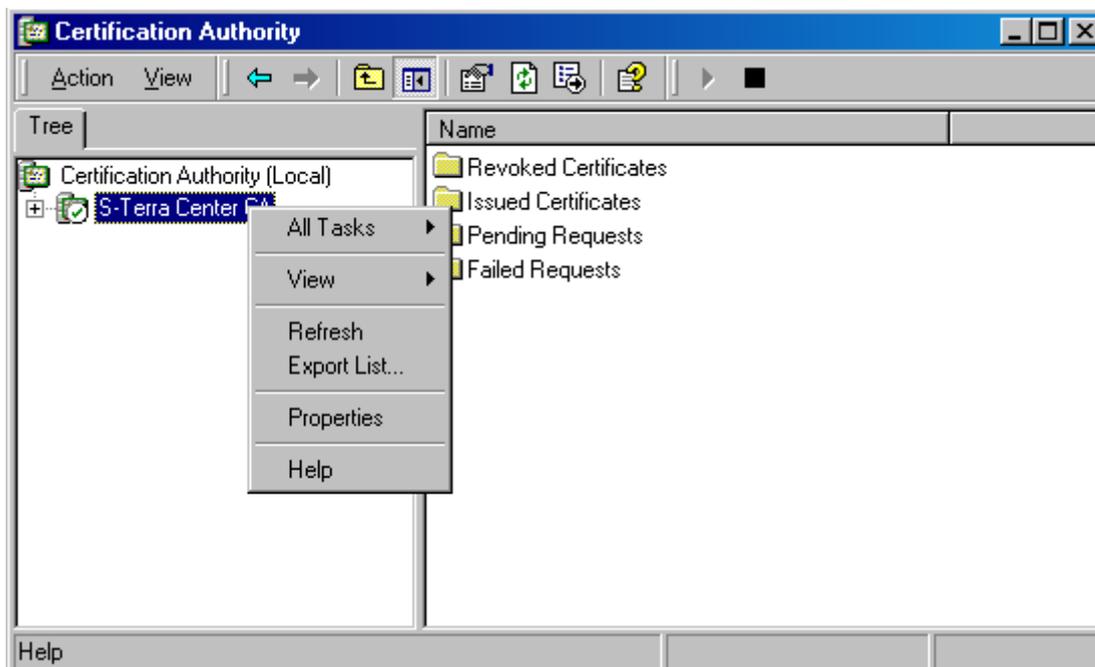


Рисунок 59

**Шаг19:** войдите в вкладку Policy Module и нажмите кнопку Configure...

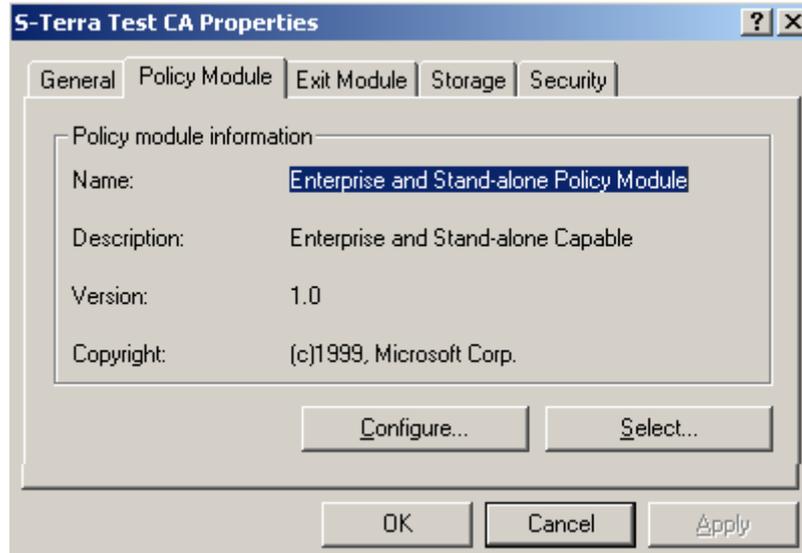


Рисунок 60

**Шаг20:** во вкладке Default Action установите переключатель в положение Always issue the certificate (всегда издавать сертификат) и нажмите ОК:

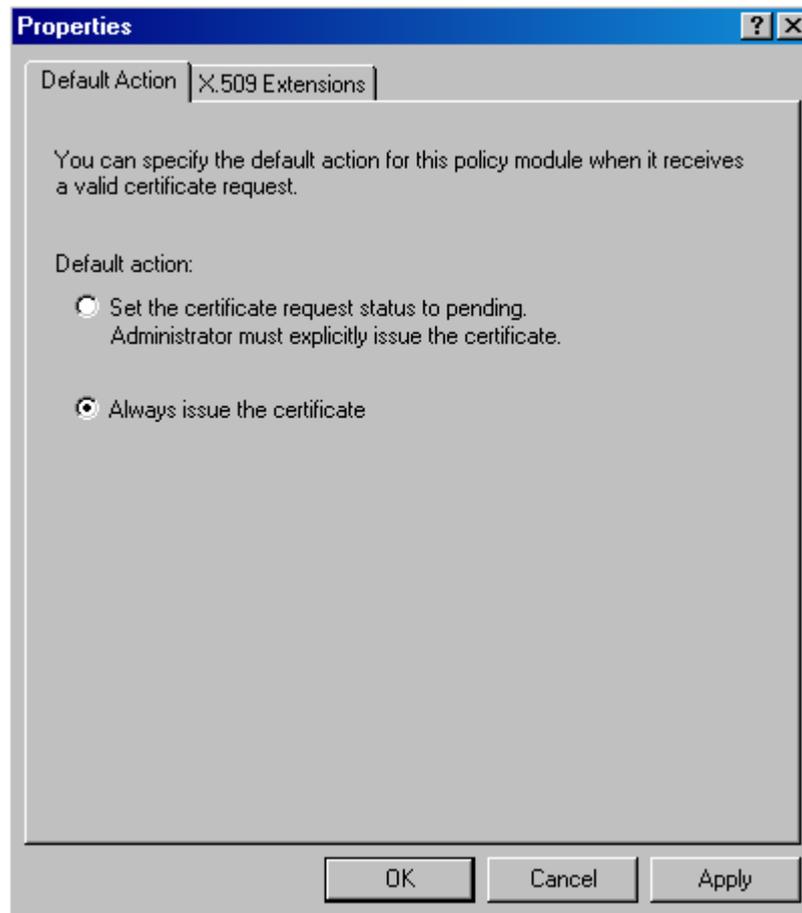


Рисунок 61

На этом создание Удостоверяющего Центра и его CA сертификата закончено.

**Примечание:**

Если была установлена версия 3.0 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.

## 10.4.4. Создание ключевой пары и запроса на локальный сертификат с помощью Microsoft Certificate Services

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 Registry.

**Шаг1:** установите программный Продукт СКЗИ “КриптоПро CSP 3.0”. Установка этого Продукта описана в разделе [“Установка СКЗИ “КриптоПро CSP”](#).

**Шаг2:** инсталлируйте ключевой носитель, на котором будет размещен контейнер с секретным ключом локального сертификата, например Registry, используя СКЗИ “КриптоПро CSP 3.0”. Эта инсталляция описана в разделе [“Инсталляция ключевого носителя Registry”](#). Для размещения контейнера на диске такая инсталляция не нужна.

**Шаг3:** запустите Microsoft Internet Explorer. В поле Address укажите IP-адрес сервера Удостоверяющего Центра и запустите утилиту `certsrv` (Certificate Service), например, `http://10.0.232.15/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 3.0”.

**Шаг4:** в появившемся окне высвечивается имя Удостоверяющего Центра – в нашем случае S-Terra Center CA. Для формирования запроса на создание локального сертификата поставьте переключатель в положение `Request a certificate` и нажмите кнопку `Next` (Рисунок 62) :

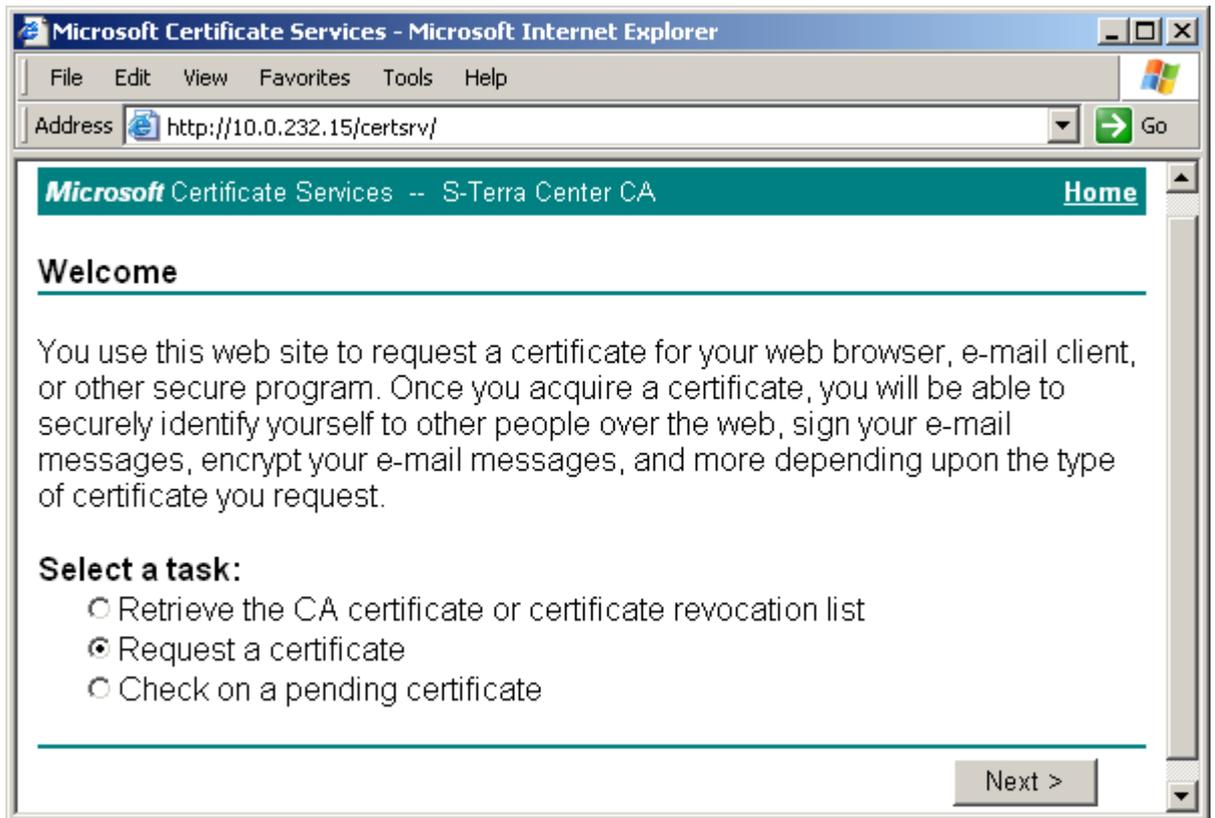


Рисунок 62

**Шаг 5:** выберите форму расширенного запроса – поставьте переключатель в положение Advanced request и наж Next:

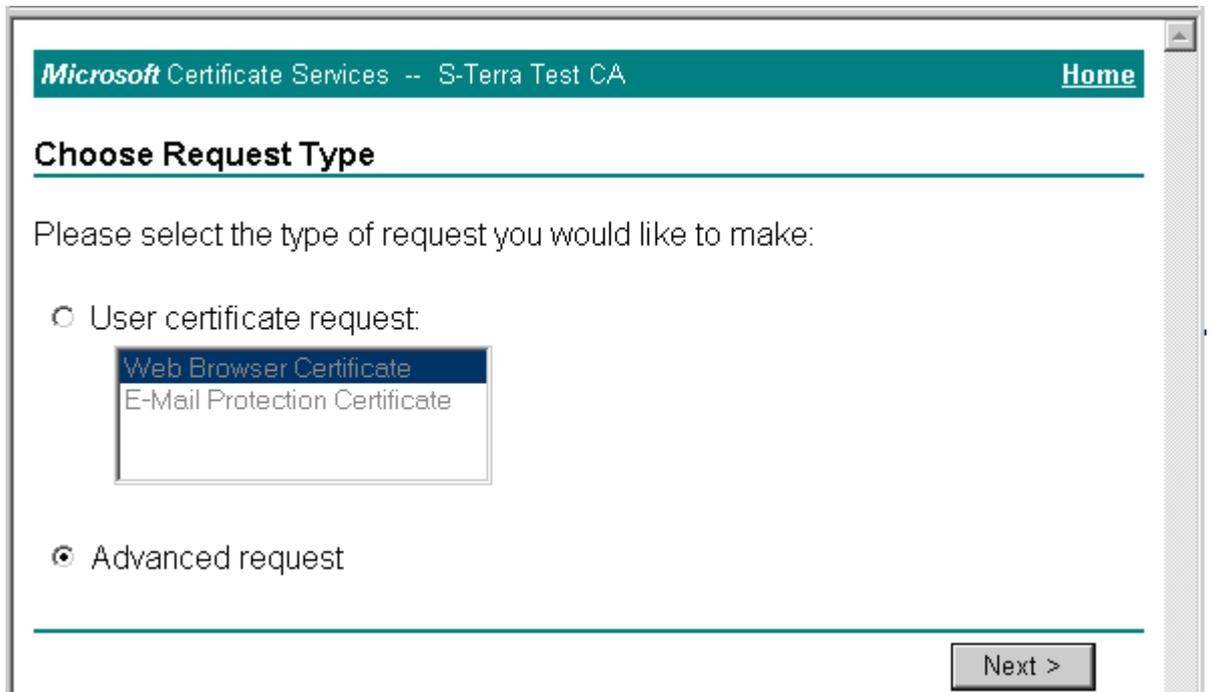


Рисунок 63

**Шаг 6:** для создания ключей и формирования запроса на сертификат воспользуемся формой, которую получим далее. Для этого - поставьте переключатель в первое положение и нажмите Next:

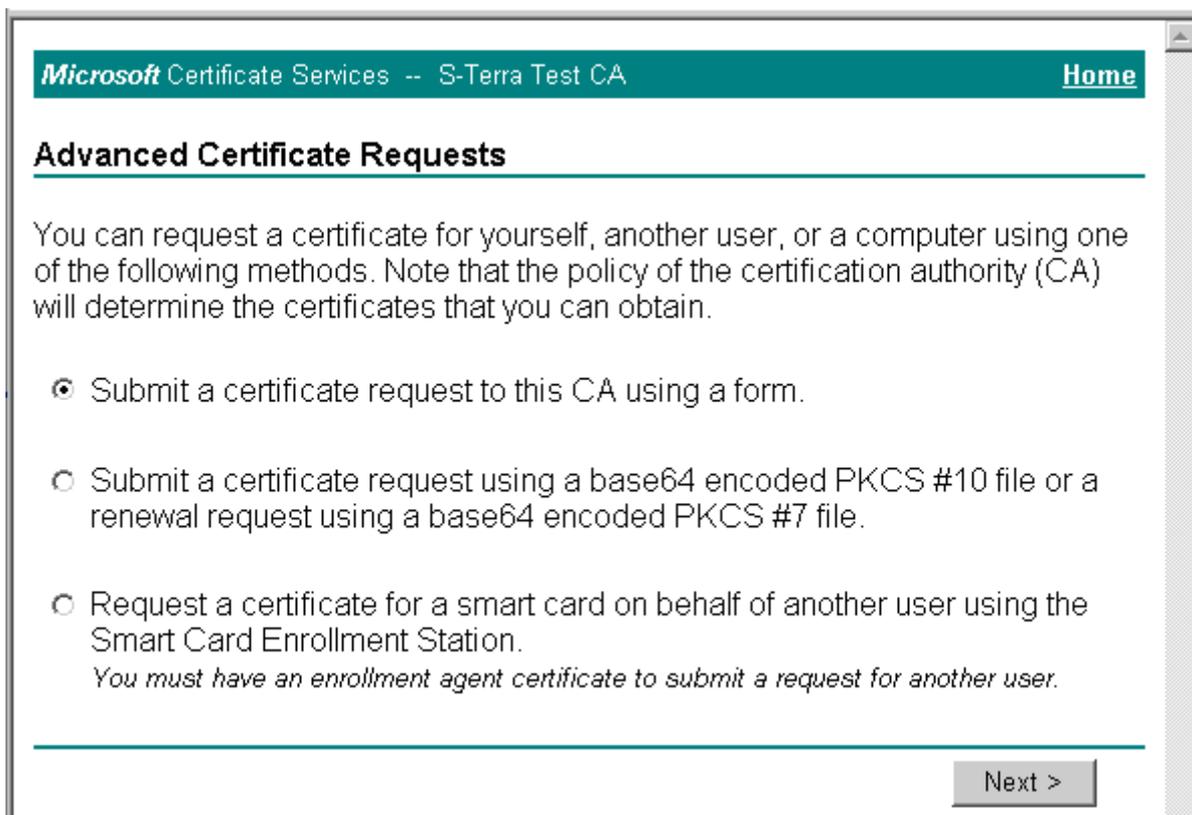


Рисунок 64

**Шаг 7:** заполните форму расширенного запроса, показанную ниже (Рисунок 65). 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.
- в разделе Intended Purpose (Область применения 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. Нужно сделать следующий выбор:
  - 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 Create new key set (Создать установки для нового секретного ключа):
    - напротив предложения Set the container name (Установить имя контейнера) поставьте флажок

- 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.
- При установке переключателя в положение Use existing key set не будет происходить создание ключевой пары, а будет использоваться уже существующий контейнер с секретным ключом:
  - в поле Container name введите имя контейнера с уже созданным ранее секретным ключом
- Enable strong private key protection - напротив этого предложения не выставлять флажок
- 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
- Use local machine 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:

E-Mail:

Company:

Department:

City:

State:

Country/Region:

---

**Intended Purpose:**

---

**Key Options:**

CSP:

Key Usage:  Exchange  Signature  Both

Key Size:  Min:512  
Max:512 (common key sizes: 512)

Create new key set

Set the container name

Container name:

Use existing key set

Enable strong private key protection

Mark keys as exportable

Export keys to file

Use local machine store

*You must be an administrator to generate a key in the local machine store.*

---

**Additional Options:**

Hash Algorithm:

*Only used to sign request.*

Save request to a PKCS #10 file

Attributes:

---

Рисунок 65

**Шаг 8 :** выберите ключевой носитель, например Registry, для размещения контейнера с секретным ключом и нажмите OK.

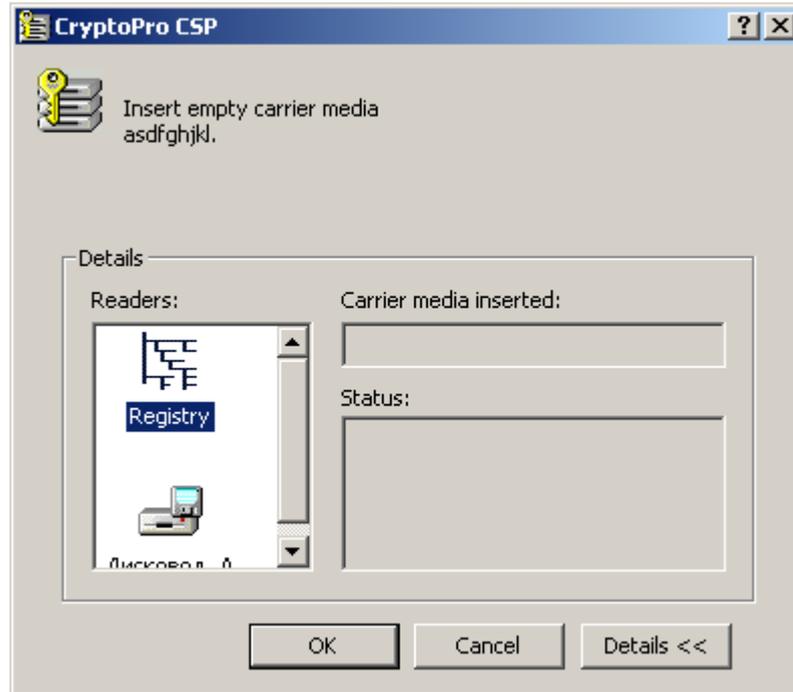


Рисунок 66

**Шаг 9 :** для создания ключевой пары генератор случайных чисел просит нажать любую клавишу или подвигать мышкой:

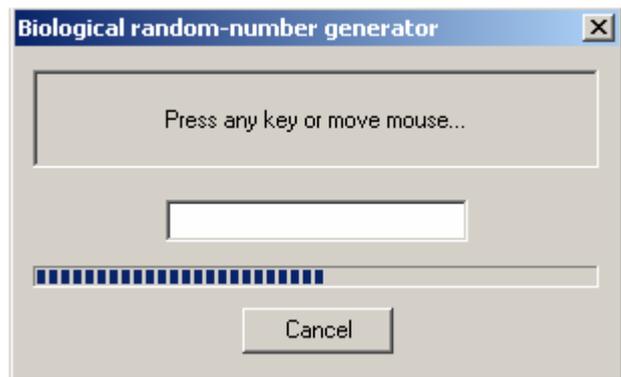


Рисунок 67

Шаг 10: задайте пароль на контейнер с секретным ключом и нажмите ОК:

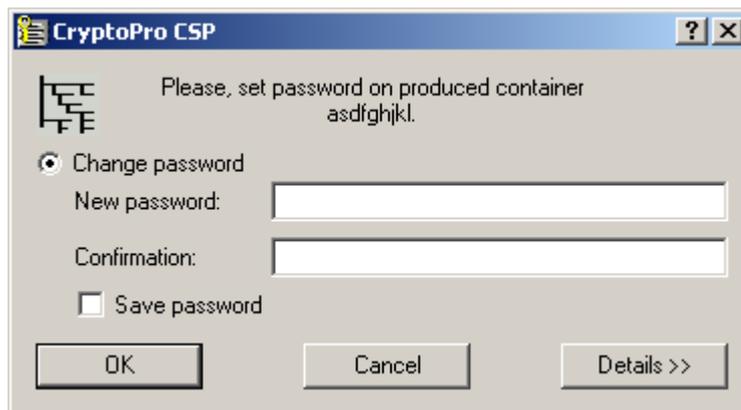


Рисунок 68

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

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 Registry.

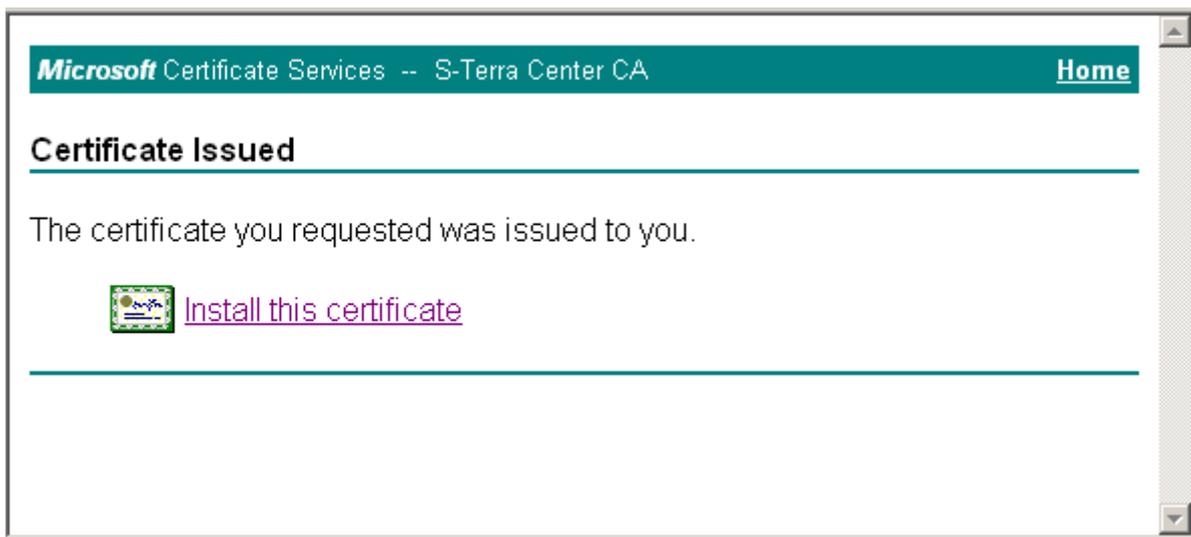


Рисунок 69

После размещения локального сертификата в контейнер выдается сообщение:

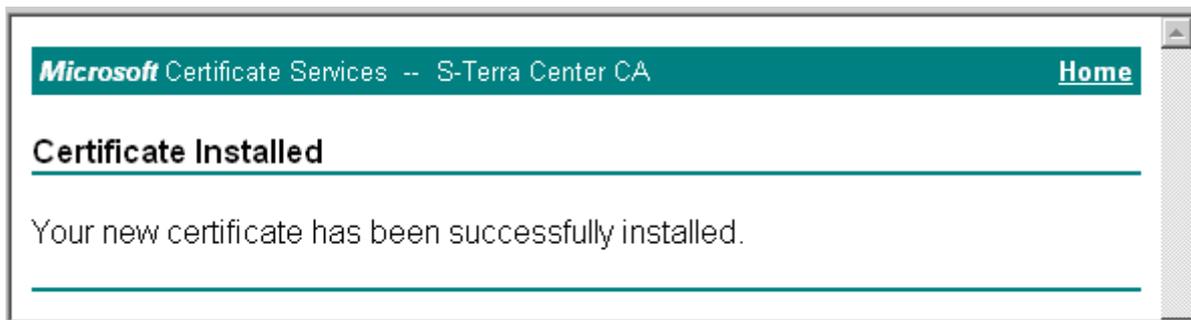


Рисунок 70

Локальный сертификат можно получить из Удостоверяющего Центра и другими путями, 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.

## 10.4.5. Экспортирование локального сертификата в файл

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

**Шаг1** : запустите продукт "КриптоПро CSP 3.0" – Start – Settings – Control Panel – CryptoPro CSP.

**Шаг2** : войдите во вкладку Service и нажмите кнопку View certificates in container ...

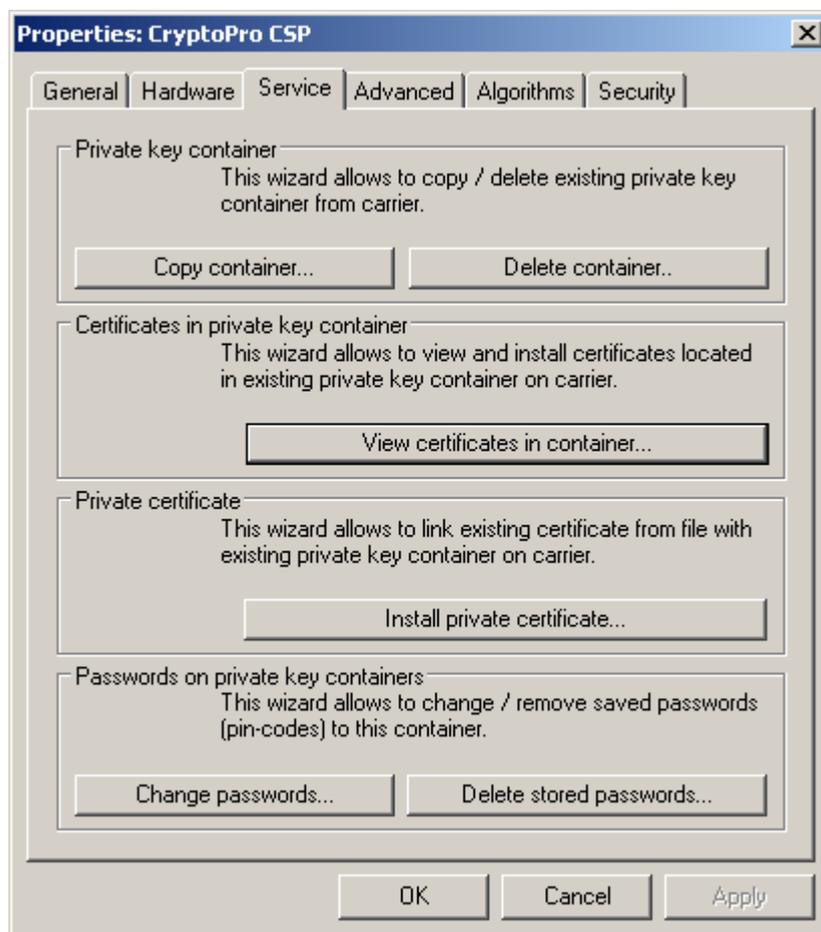


Рисунок 71

**Шаг 3** : в следующем окне для указания контейнера поставьте переключатель в положение Computer и нажмите кнопку Browse...

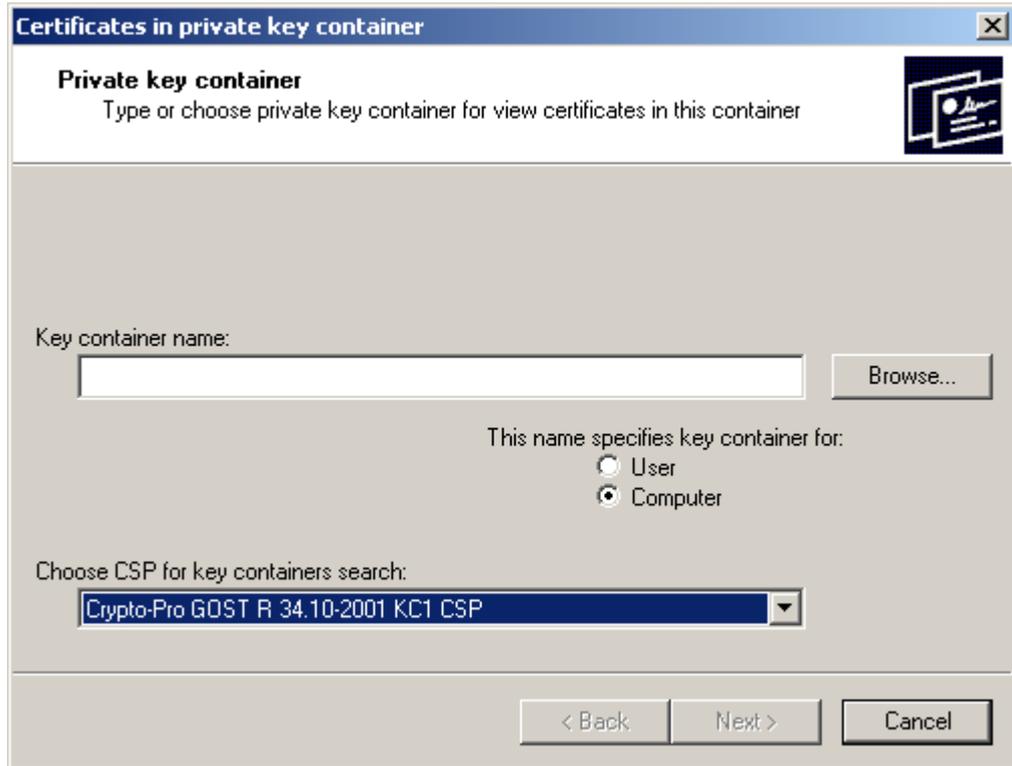


Рисунок 72

**Шаг 4** : в окне Select key container выберите имя контейнера, в котором лежит секретный ключ локального сертификата и сам сертификат, и поставьте переключатель в положение Unique names, и нажмите кнопку ОК:



Рисунок 73

Шаг 5: выбор контейнера произведен нажмите кнопку Next:

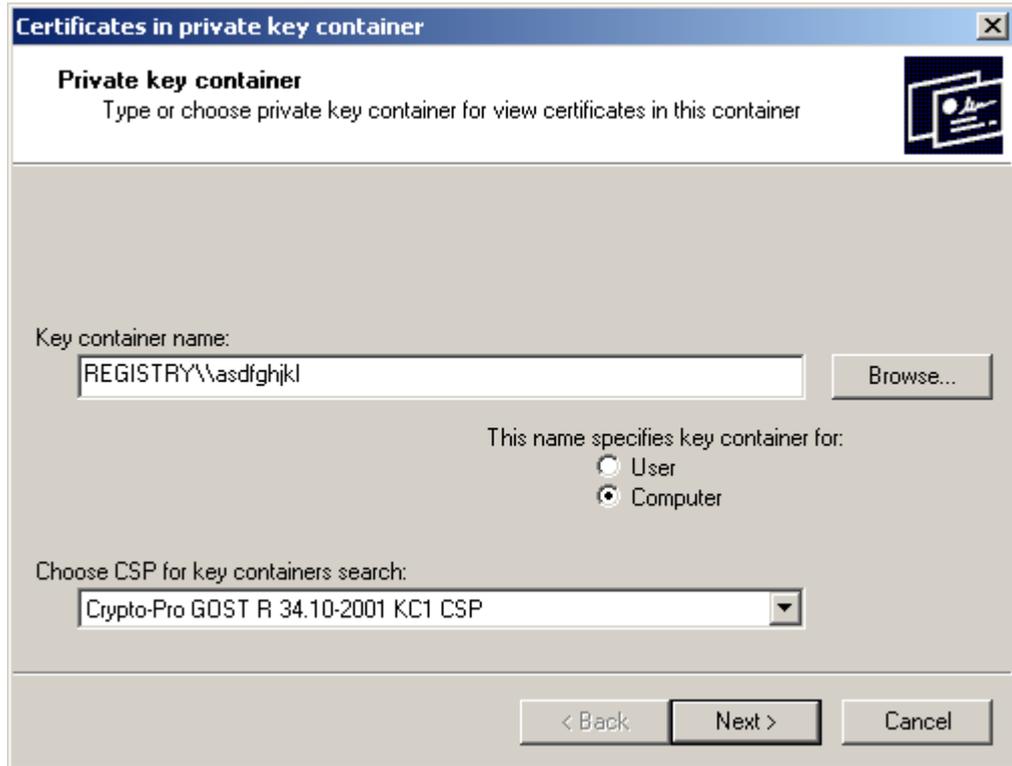


Рисунок 74

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

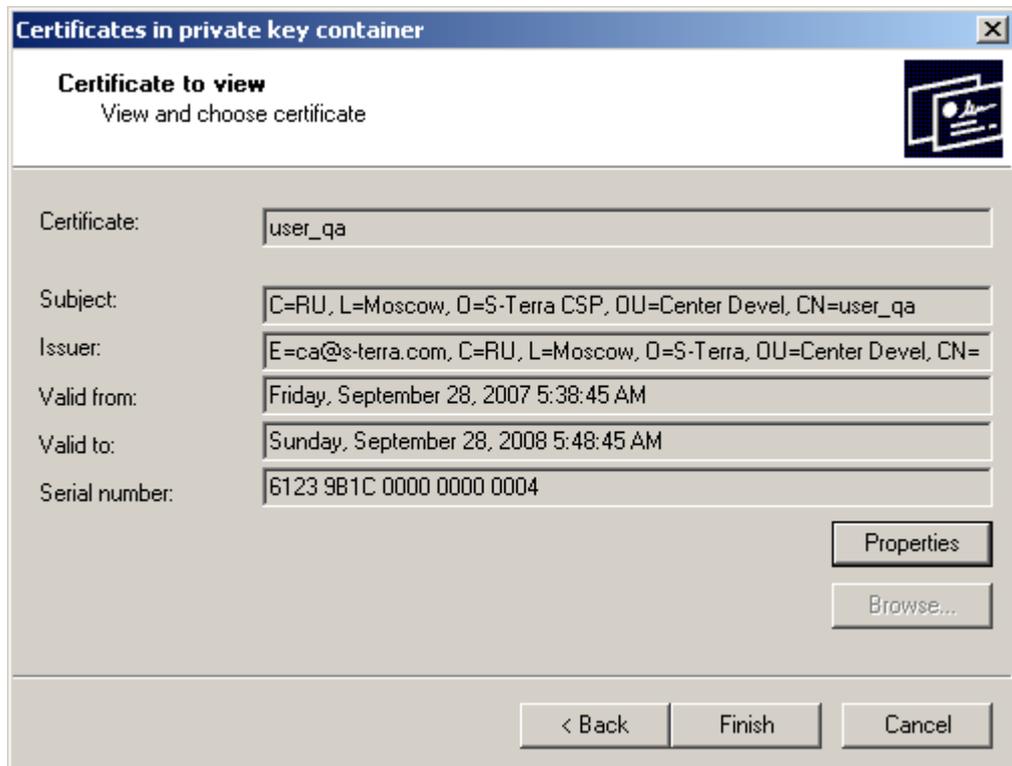


Рисунок 75

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

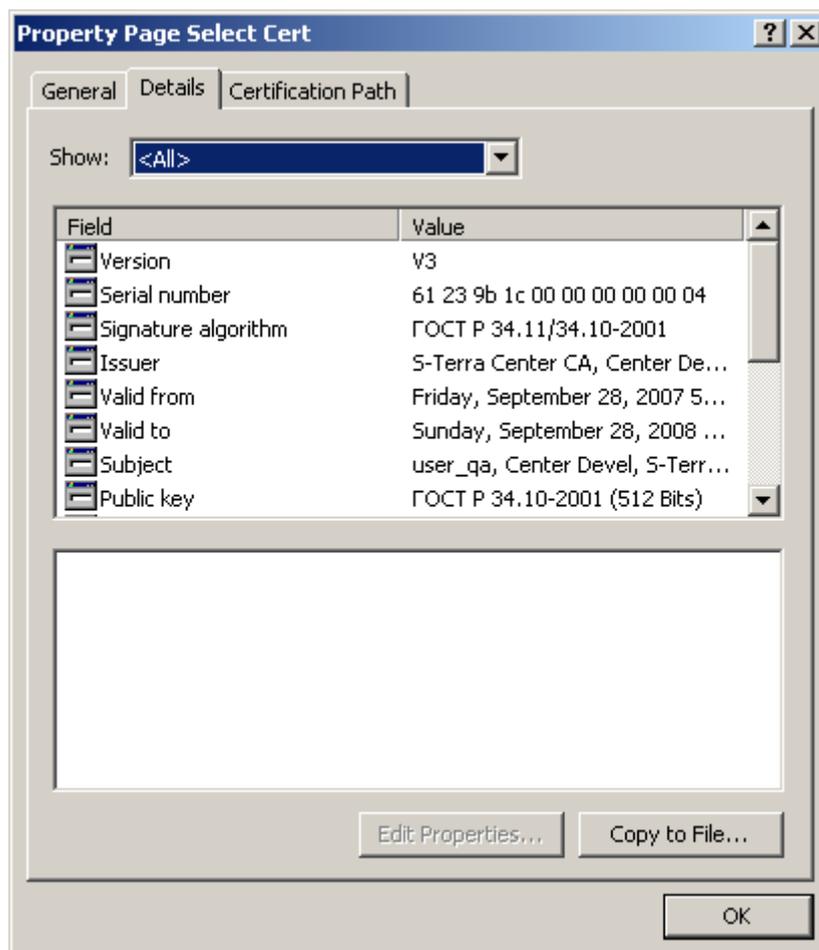


Рисунок 76

Шаг 8: в окне визарда нажмите кнопку Next:



Рисунок 77

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



Рисунок 78

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

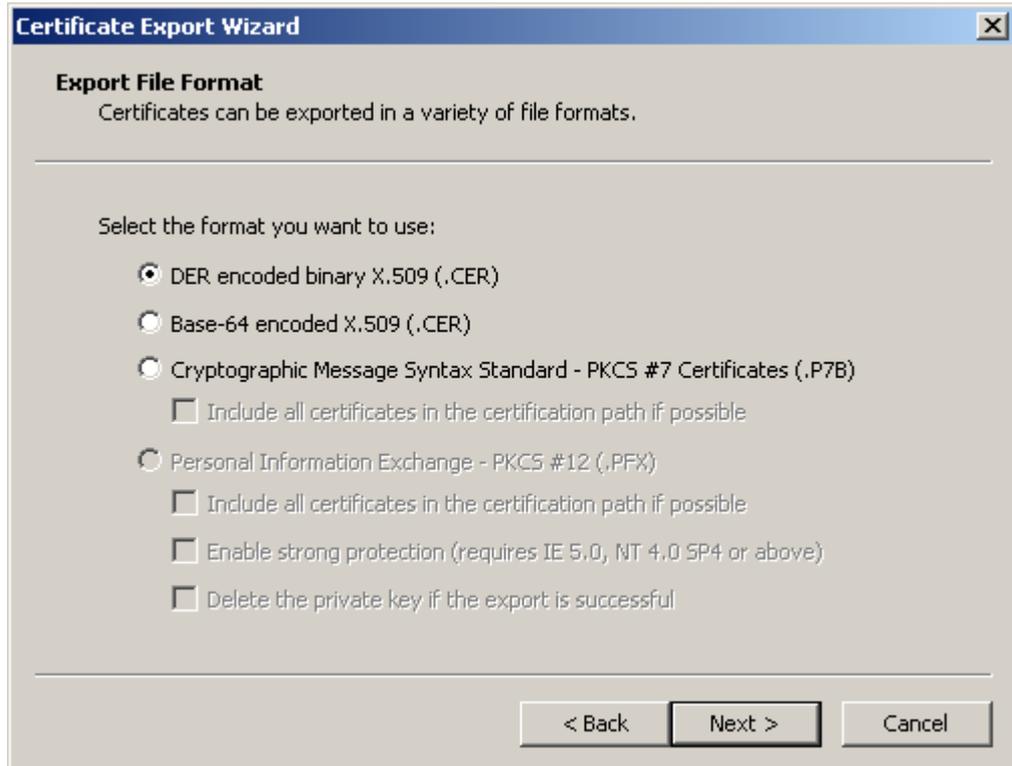


Рисунок 79

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

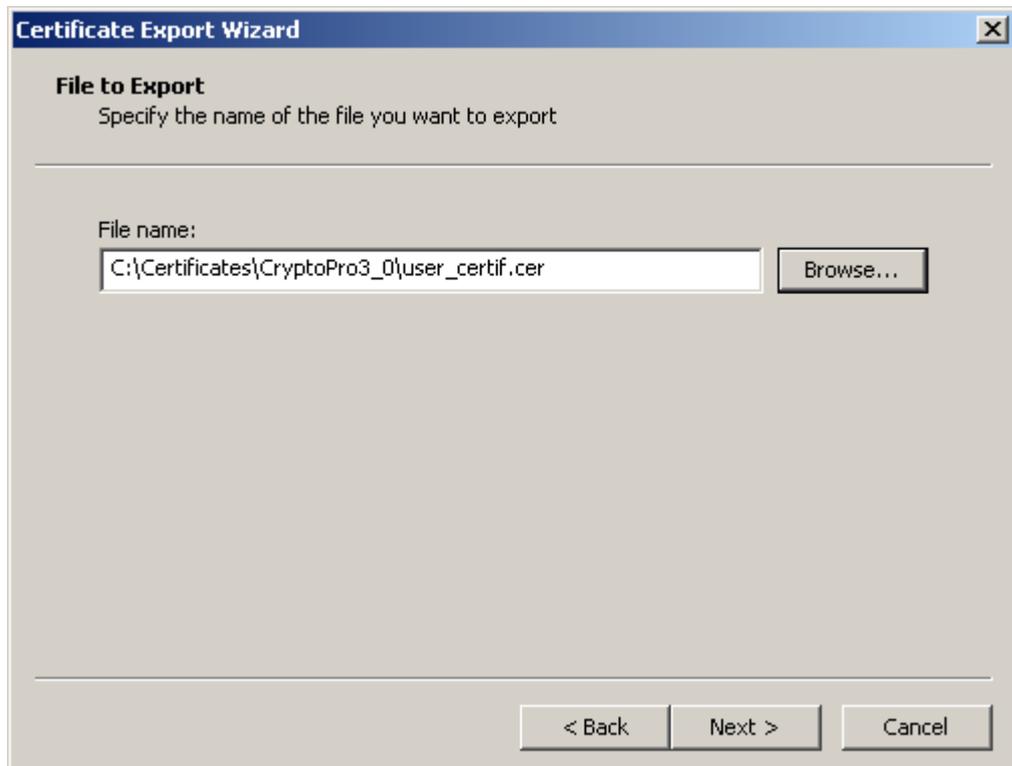


Рисунок 80

**Шаг12:** экспортирование локального сертификата в файл закончено, нажмите Finish.



Рисунок 81

На этом создание локального сертификата для шлюза безопасности и СА 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 для доставки на шлюз безопасности создать на нем виртуальный дисковод, что и описано в разделе ["Создание виртуального дисковода"](#) в данном Приложении.

## 10.5. Создание локального сертификата в "КриптоПро CSP 2.0"

### 10.5.1. Инсталляция ключевого носителя Registry в "КриптоПро CSP 2.0"

Для инсталляции локального ключевого носителя Registry надо выполнить следующие действия:

**Шаг1:** запустить КриптоПро CSP: `Start -Settings-Control Panel - CryptoPro CSP`

**Шаг2:** в появившемся окне Properties войти во вкладку Hardware и нажать кнопку Configure carries (настроить носители) (Рисунок 82).

**Шаг3:** нажать кнопку Add, чтобы добавить новый ключевой носитель (Рисунок 83).

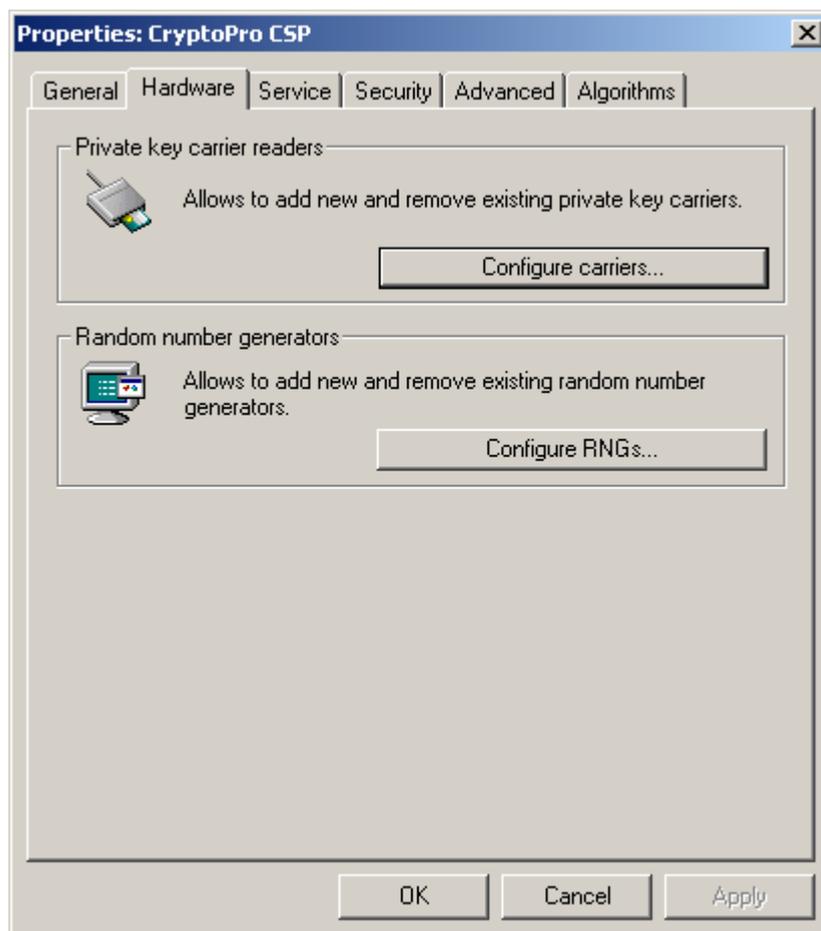


Рисунок 82

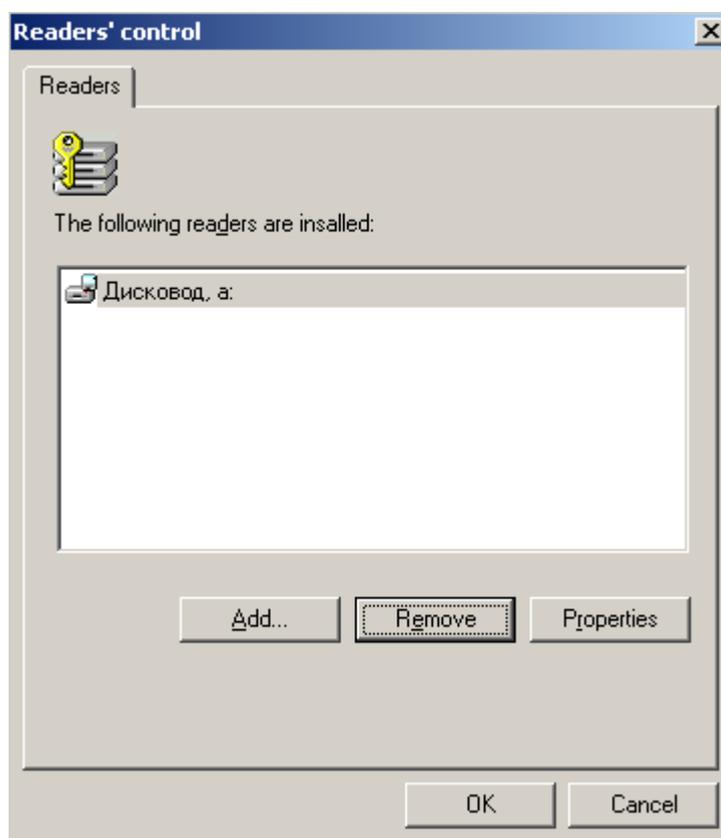


Рисунок 83

**Шаг 4 :** выбрать новый носитель Registry и нажать Next :

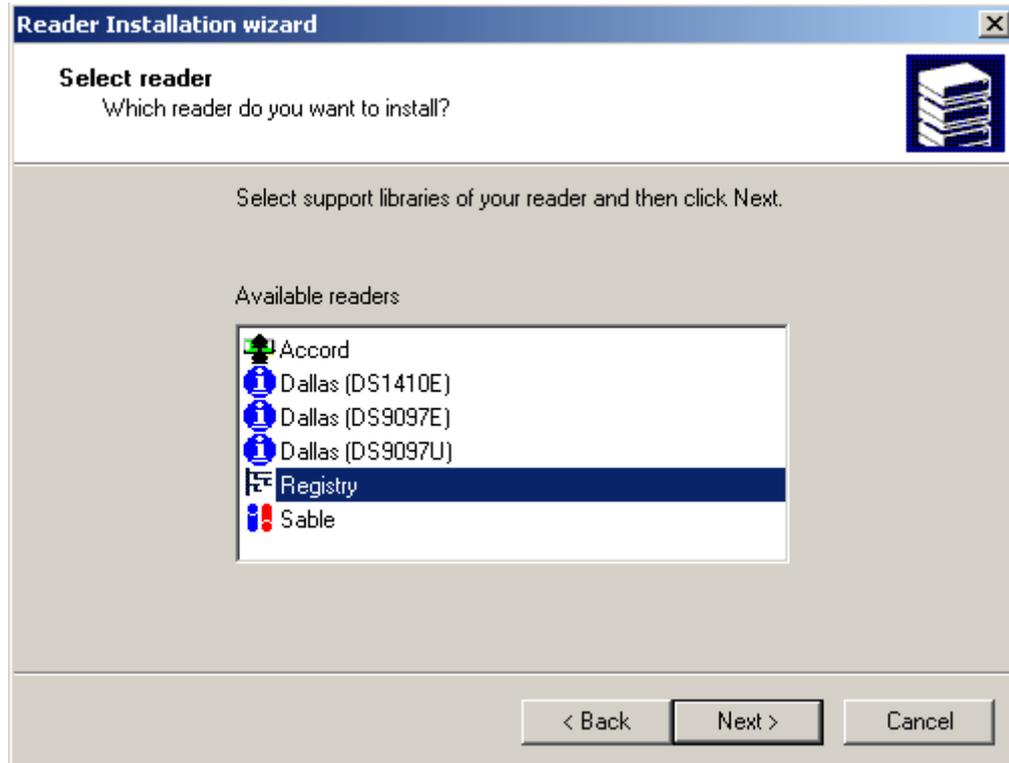


Рисунок 84

**Шаг 5 :** новый ключевой носитель Registry инсталлирован, нажать OK :



Рисунок 85

## 10.5.2. Установка внешнего считывателя ключевой информации в "КриптоПро CSP 2.0"

Для установки внешнего считывателя ключевой информации, например eToken, выполните следующие действия:

**Шаг1** : подключить внешний считыватель к компьютеру пользователя (eToken до установки драйверов подключать не следует).

**Шаг2** : установить драйвер этого устройства и драйвер для взаимодействия устройства с "КриптоПро CSP". См. раздел ["Подключение внешних ключевых считывателей \(носителей\)"](#).

**Шаг3** : запустить "КриптоПро CSP": ( Start -Settings-Control Panel - CryptoPro CSP)

**Шаг4** : в появившемся окне Properties войти во вкладку Hardware и нажать кнопку Configure carries (настроить носители):

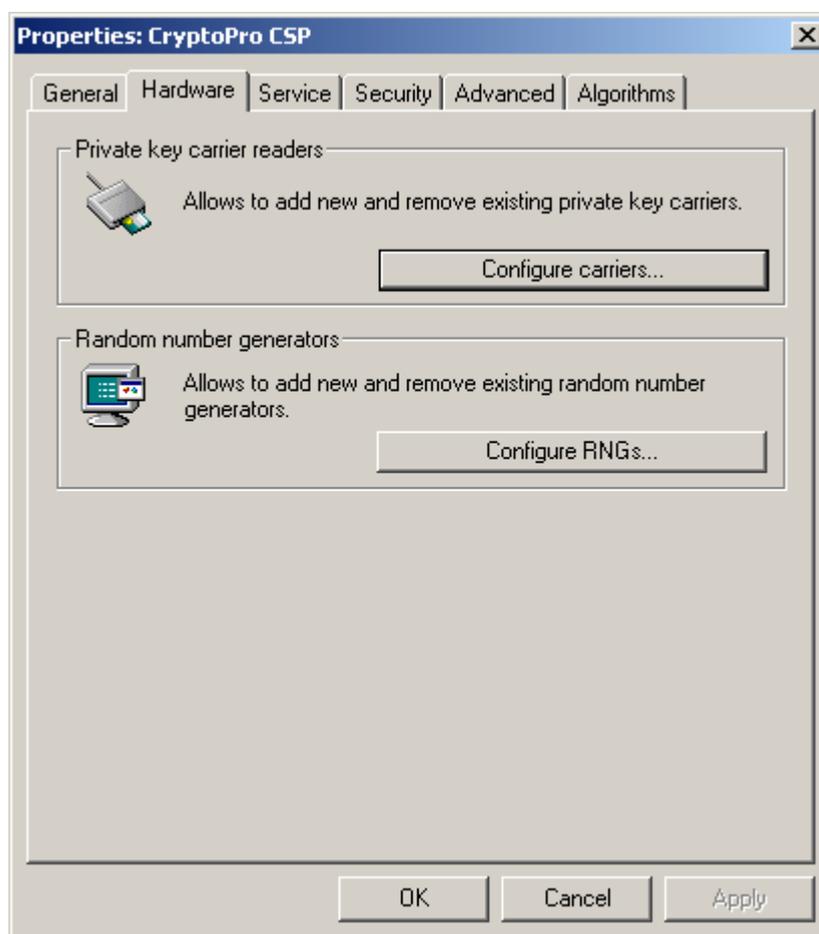


Рисунок 86

**Шаг 5:** для инсталляции нового считывателя нажать кнопку Add:

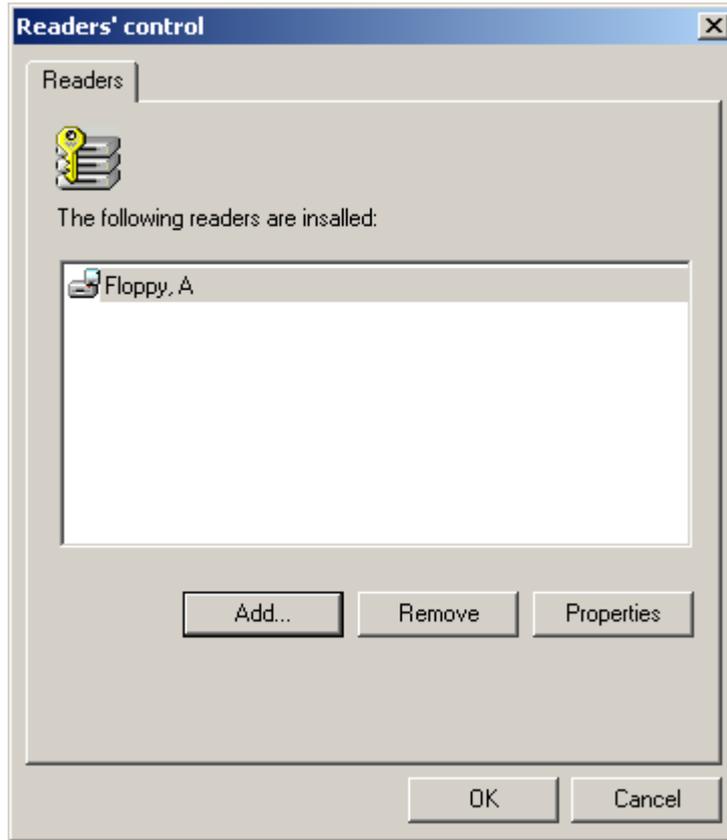


Рисунок 87

**Шаг 6:** выбрать новый считыватель, например, для eToken – AKS ifdh 0, и нажать Next:

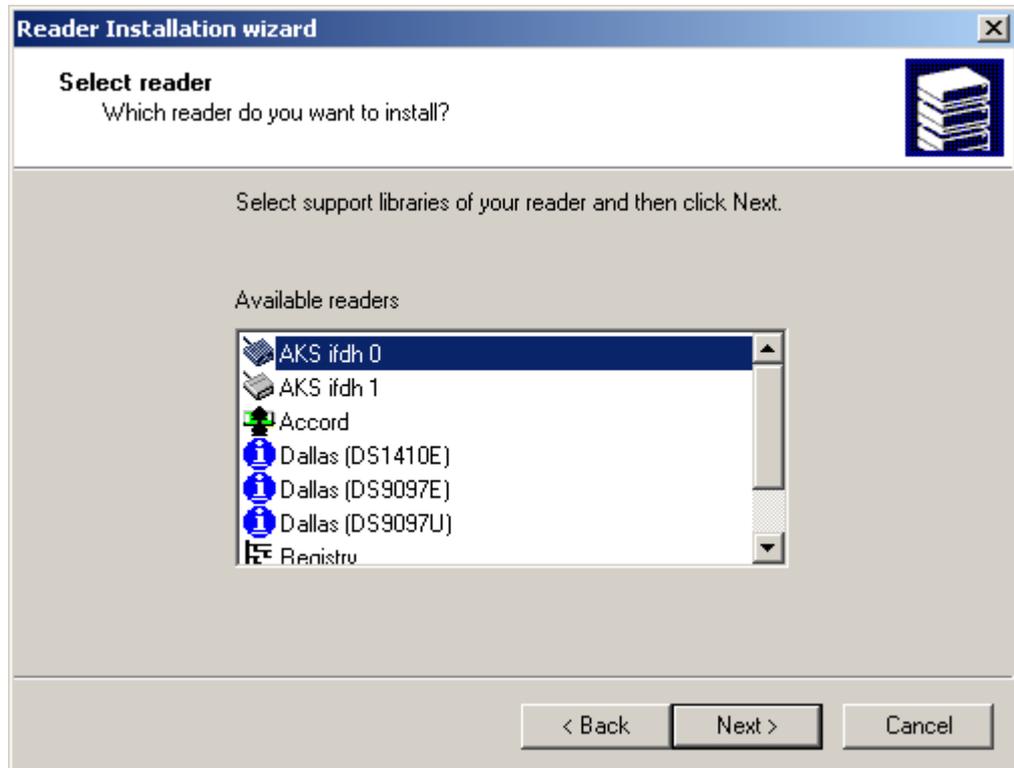


Рисунок 88

Шаг 7 : можно присвоить имя этому считывателю и нажать Next :

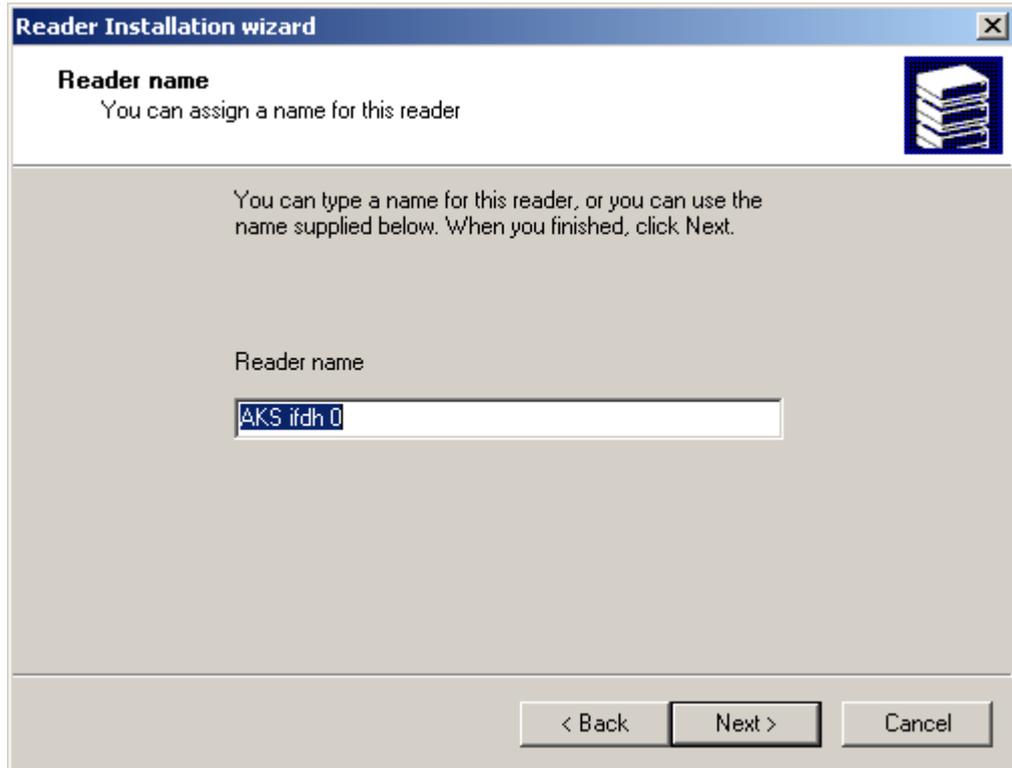


Рисунок 89

Шаг 8 : инсталляция нового считывателя eToken завершена, нажмите Finish:

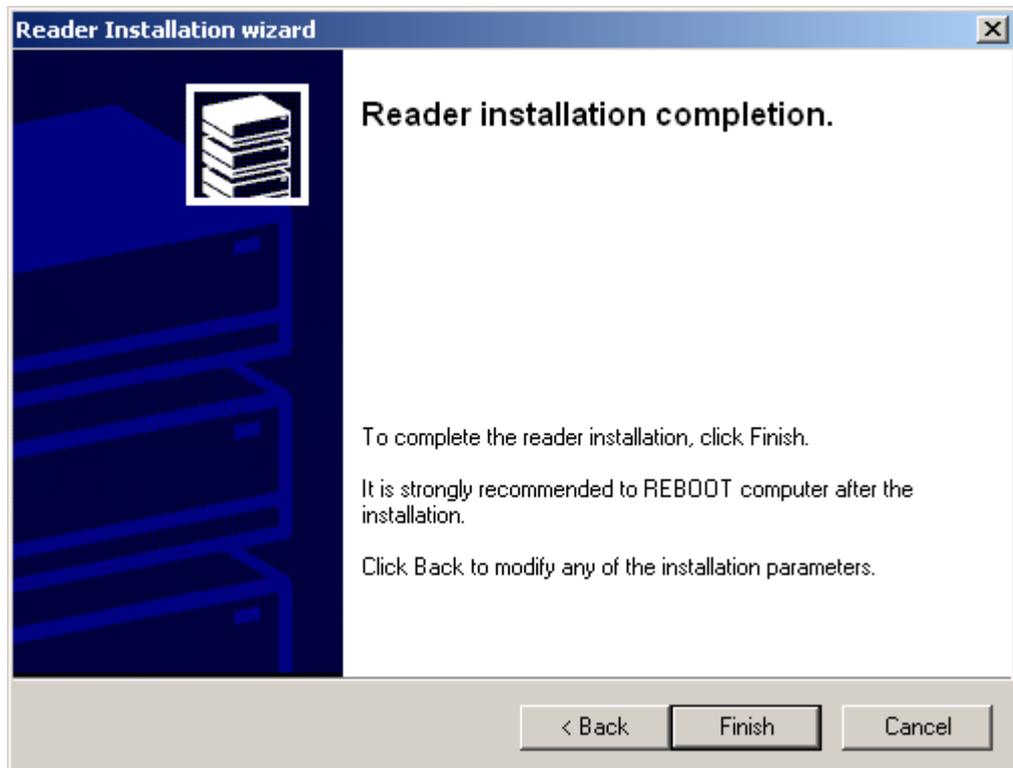


Рисунок 90

**Шаг 9:** считыватель AKS ifdh 0 для eToken добавлен в список установленных считывателей, нажмите ОК:



Рисунок 91

**Замечание:**

КриптоПро CSP позволяет установить два считывателя для eToken (Рисунок 92).

При этом имена контейнеров, хранящихся на eToken, не содержат имена считывателей с этих устройств, например:

```
SCARD\ETOKEN_R2_00070188\CC00\B2E1
```

```
SCARD\ETOKEN_R2_00070188\CC00\B2E1
```

И при попытке получить доступ к контейнеру по имени, функция доступа возвращает ошибку. Вследствие этого продукт CSP VPN Client не может "добраться" до контейнера и выдается сообщение, что контейнер с заданным именем не найден.

Для предотвращения подобной ситуации, необходимо установить только один считыватель для токенов, например AKS ifdh 0.

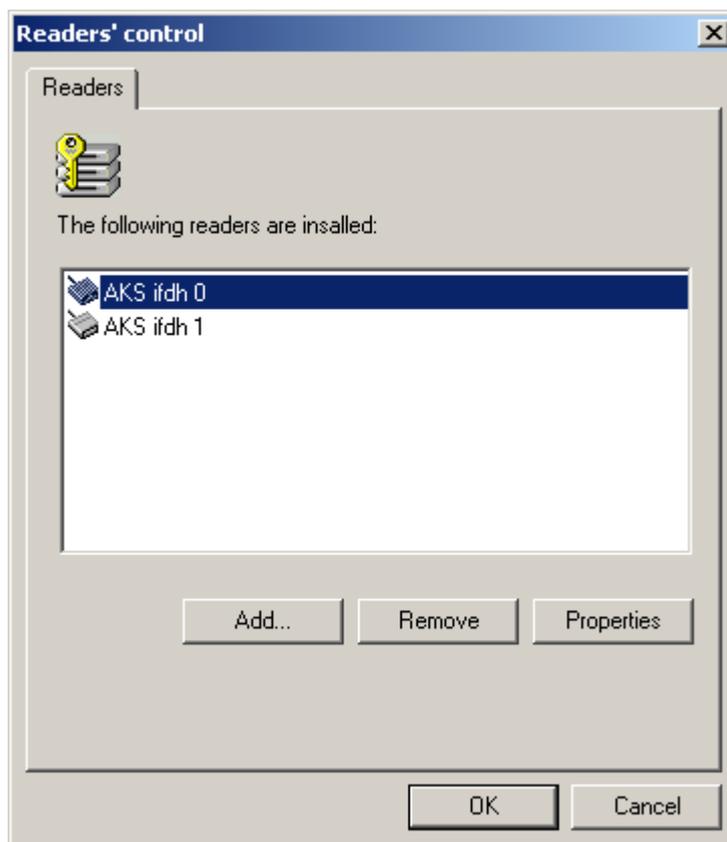


Рисунок 92

### 10.5.3. Установка и настройка Удостоверяющего Центра. Создание СА сертификата

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

На отдельном компьютере установите ОС Windows 2000 (2003) Server и СКЗИ «КриптоПро CSP 2.0». Сервис Internet Information Services (IIS) должен быть включен. Все дальнейшие действия выполняются также, как описано для «КриптоПро CSP 3.0» в разделе [«Установка и настройка Удостоверяющего Центра. Создание СА сертификата»](#).

### 10.5.4. Создание ключевой пары и запроса на локальный сертификат

Для создания ключевой пары и формирования запроса на создание локального сертификата можно использовать средства Microsoft Windows. Этот процесс описан для «КриптоПро CSP 3.0» в разделе [«Создание ключевой пары и запроса на локальный сертификат»](#).

После отсылки запроса на сертификат вместо окна с предложением установить локальный сертификат появляется окно с уведомлением о получении запроса на сертификат (Рисунок 93).

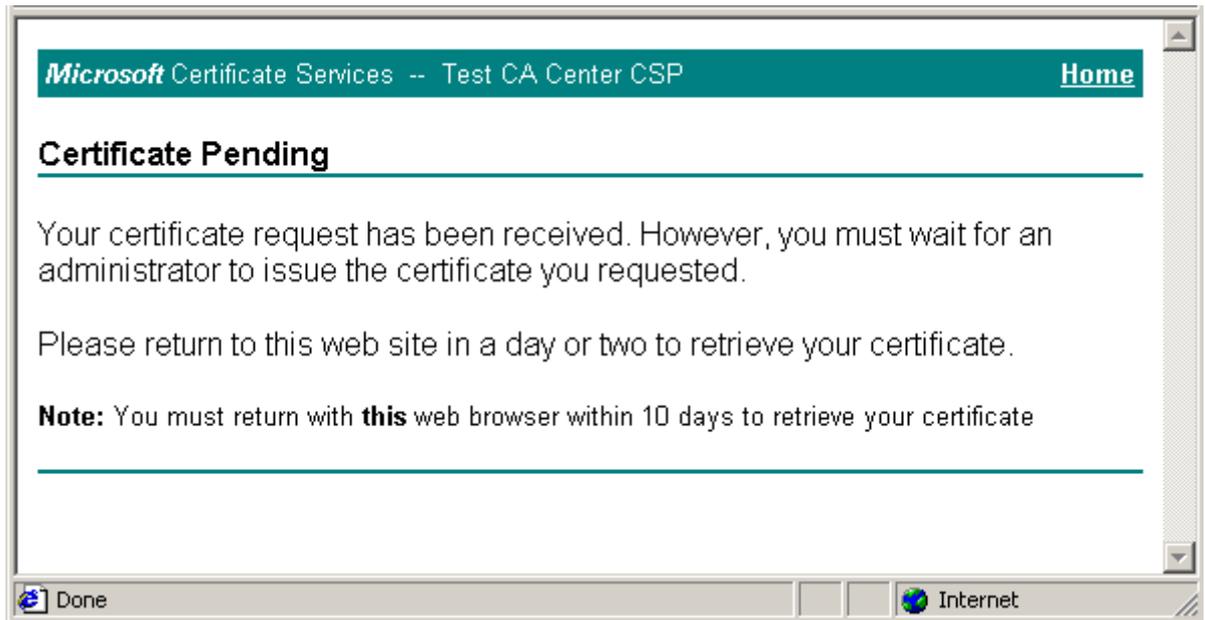


Рисунок 93

Если вы создали Удостоверяющий Центр, то в следующем разделе описано, как создать локальный сертификат по полученному запросу и экспортировать его в файл.

## 10.5.5. Создание локального сертификата с "КриптоПро CSP 2.0"

Описываем создание локального сертификата по полученному запросу также средствами Microsoft Windows. На сервере Удостоверяющего Центра, на котором должен быть установлен Windows 2000 Server, Certificate Services, СКЗИ "КриптоПро CSP 2.0", администратор создает локальный сертификат.

**Шаг 1:** войти в Удостоверяющий Центр (Start - Settings - Control Panel - Administrative Tools - Certification Authority), откроется окно Certification Authority:

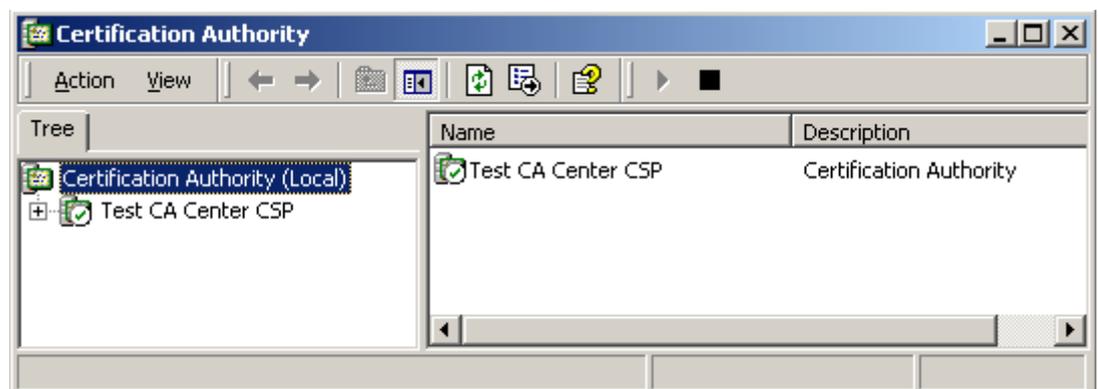


Рисунок 94

**Шаг 2:** при выборе названия Удостоверяющего Центра ( в нашем случае Test CA Center CSP ) откроется список папок с разными сертификатами:



Рисунок 95

**Шаг 3:** при выборе папки Pending Request (Запросы на создание сертификатов) появляется список запросов. Выделить курсором запрос, по которому надо создать сертификат:

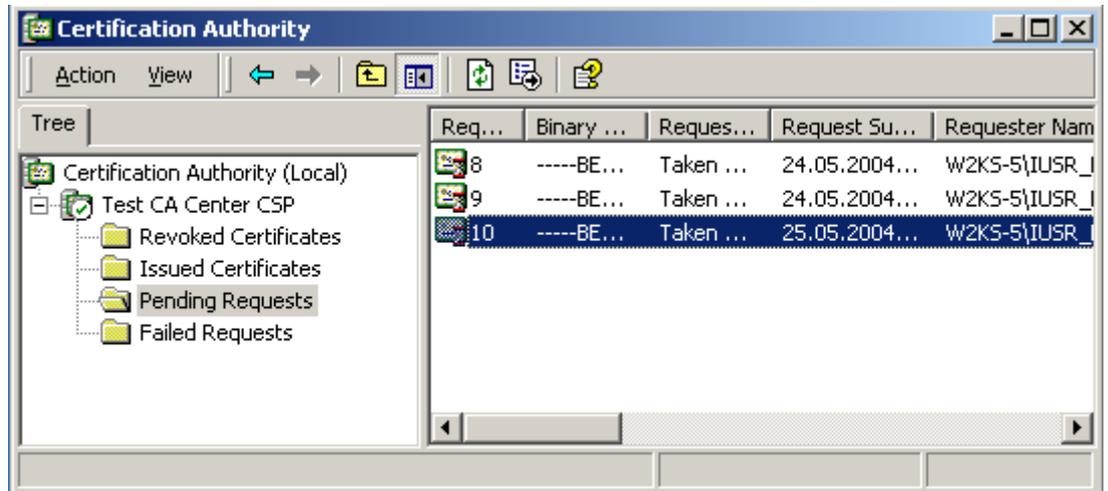


Рисунок 96

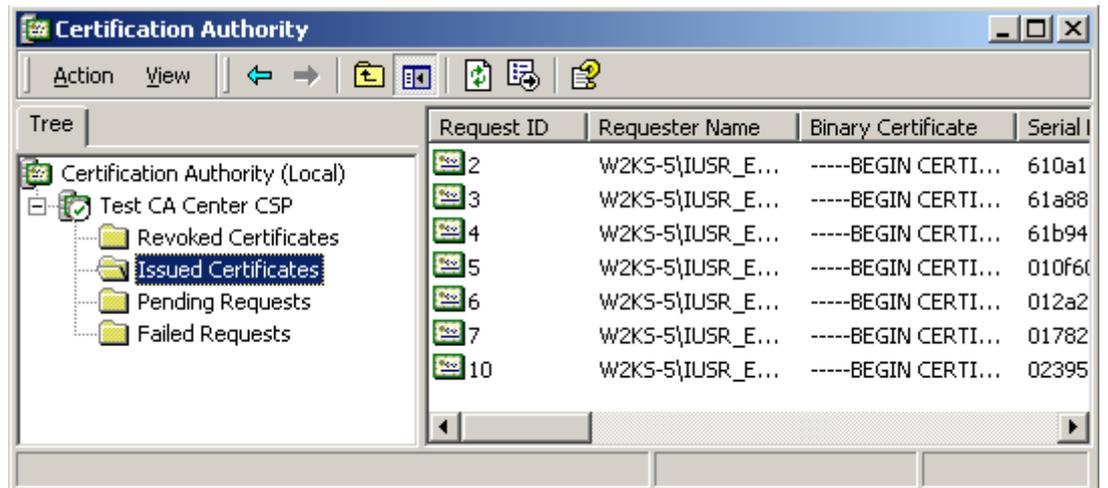


Рисунок 97

**Шаг 4 :** в меню Action выбрать команду All Tasks, затем Issue для создания сертификата. Созданный сертификат появится в папке созданных сертификатов (Issued Certificates) (Рисунок 97).

**Шаг 5 :** для открытия созданного сертификата выделить курсором сертификат и в меню Action выбрать команду Open :

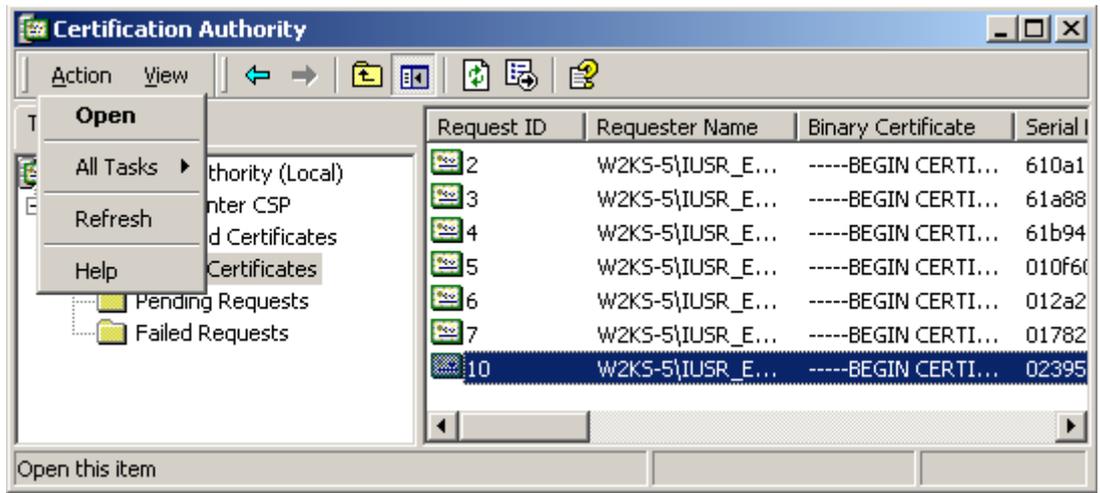


Рисунок 98

**Шаг 6 :** для экспортирования сертификата в файл выбрать вкладку Details. В выпадающем списке Show выбрать предложение All, чтобы увидеть все поля сертификата и нажать кнопку Copy to File.

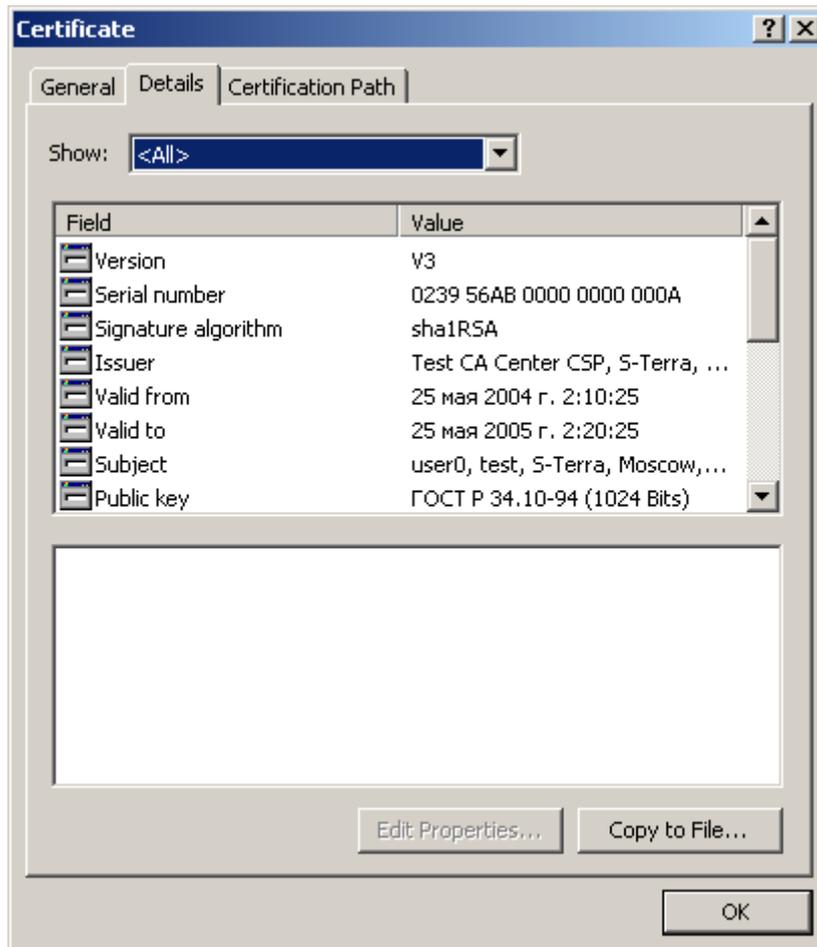


Рисунок 99

**Шаг 7:** в окне визарда начала процесса экспортирования сертификата в файл нажать Next:



Рисунок 100

**Шаг 8:** выбрать формат файла, поставив переключатель в положение DER encoded binary X.509 (.CER), и нажать Next:

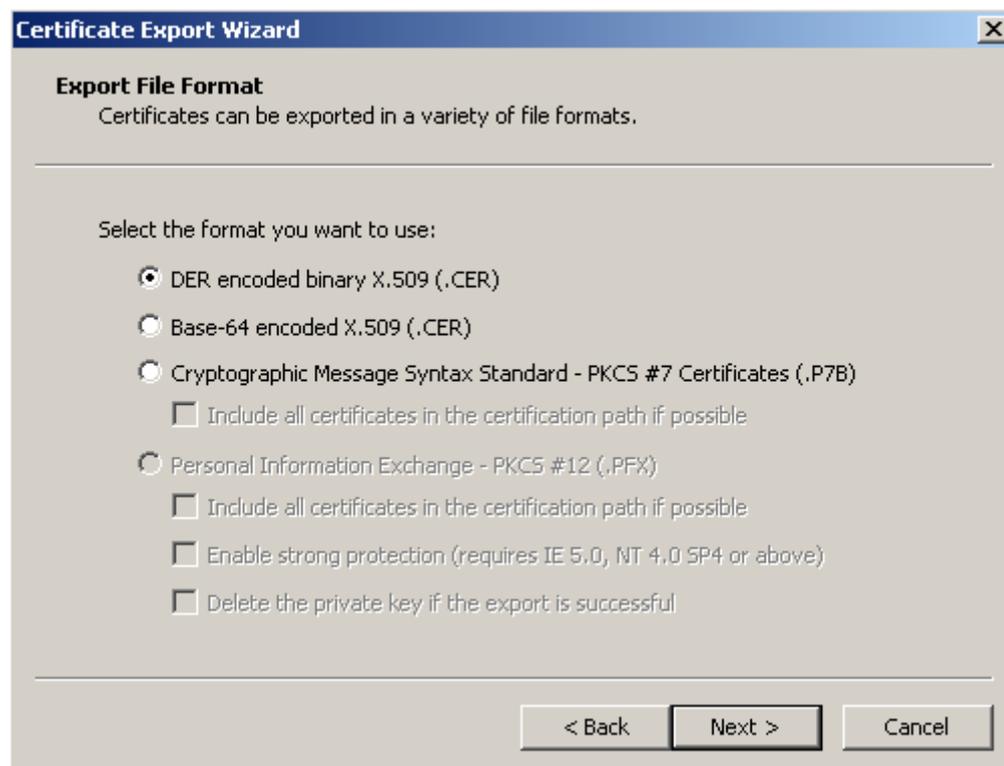


Рисунок 101

**Шаг9:** экспортируйте локальный сертификат на дискету, указав имя файла и нажмите Next:

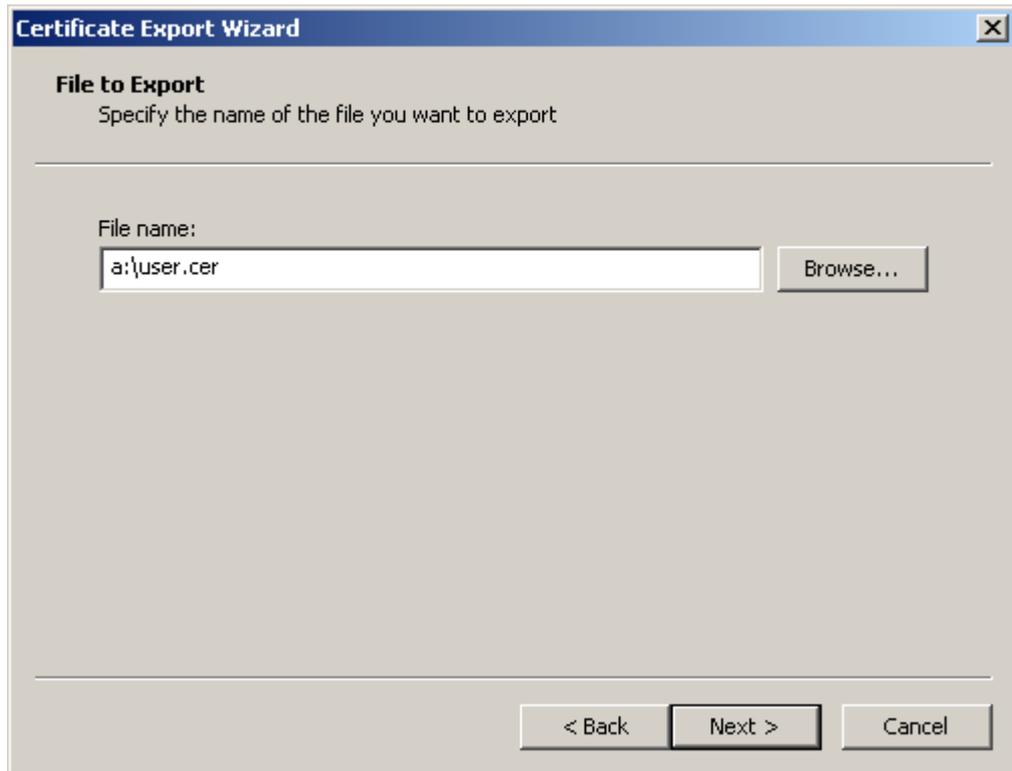


Рисунок 102

**Шаг10:** экспортирование локального сертификата в файл завершено, нажмите Finish.



Рисунок 103

Доставьте оба сертификата на шлюз безопасности любым доступным способом и зарегистрируйте их. Контейнер с секретным ключом локального сертификата можно скопировать на дискету и для доставки на шлюз безопасности создать на нем виртуальный дисковод, что и описано в разделе [“Создание виртуального дисковода”](#) в данном Приложении.

## 10.6. Создание ключевой пары и запроса на локальный сертификат с помощью утилиты `cryptcp` в ОС Red Hat Linux 9 и ОС Solaris 9

Если продукт CSP VPN Gate работает с ОС Red Hat Linux 9 или ОС Solaris 9, то в состав поставляемого образа диска входит пакет «КриптоПро CSP 3.0» с утилитой `cryptcp`, созданный компанией "Крипто-Про", и который рекомендовалось установить самостоятельно при инсталляции CSP VPN Gate. (В состав пакета «КриптоПро CSP 2.0» утилита `cryptcp` не входит).

Утилита `cryptcp` размещена в каталоге:

`/usr/CPROcsp/bin` (в ОС Red Hat Linux 9)

`/opt/CPROcsp/bin` (в ОС Solaris 9).

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

`cd /usr/CPROcsp/bin` (в ОС Red Hat Linux 9)

или

`cd /opt/CPROcsp/bin` (в ОС Solaris 9)

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

где

<code>/tmp/g100.req</code>	имя файла с запросом на сертификат в формате PKCS#10
<code>provtype 75</code>	тип криптопровайдера, по умолчанию 75 – Crypto-Pro GOST R34.10-2001 CSP ( 71 - Crypto-Pro GOST R34.10-94 CSP)
<code>HDIMAGE</code>	имя считывателя (жесткий диск)
<code>g100</code>	имя контейнера
<code>both</code>	создается два типа ключей - для подписи и шифрования
<code>km</code>	разместить контейнер на компьютере
<code>dn</code>	поля сертификата.

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 3.0”. 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”.

# 11. 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"
- установить 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.

## 11.1. 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/encrypt/signal\\_com/](http://www.signal-com.ru/ru/prod/encrypt/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:

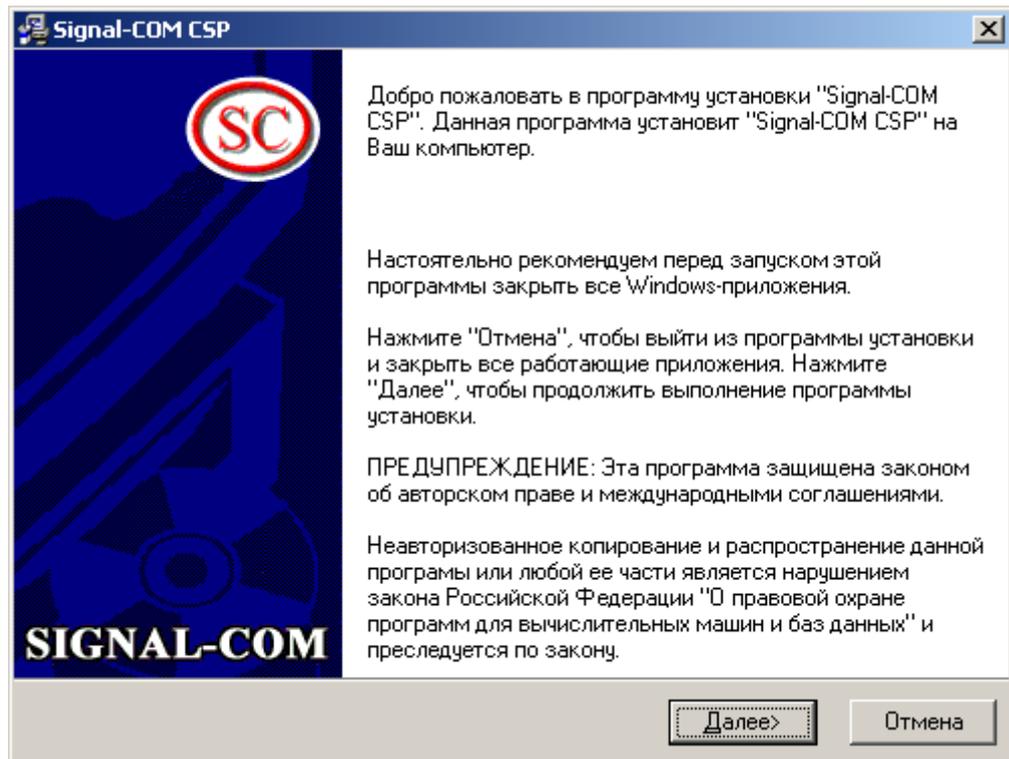


Рисунок 104

Введите требуемую информацию:

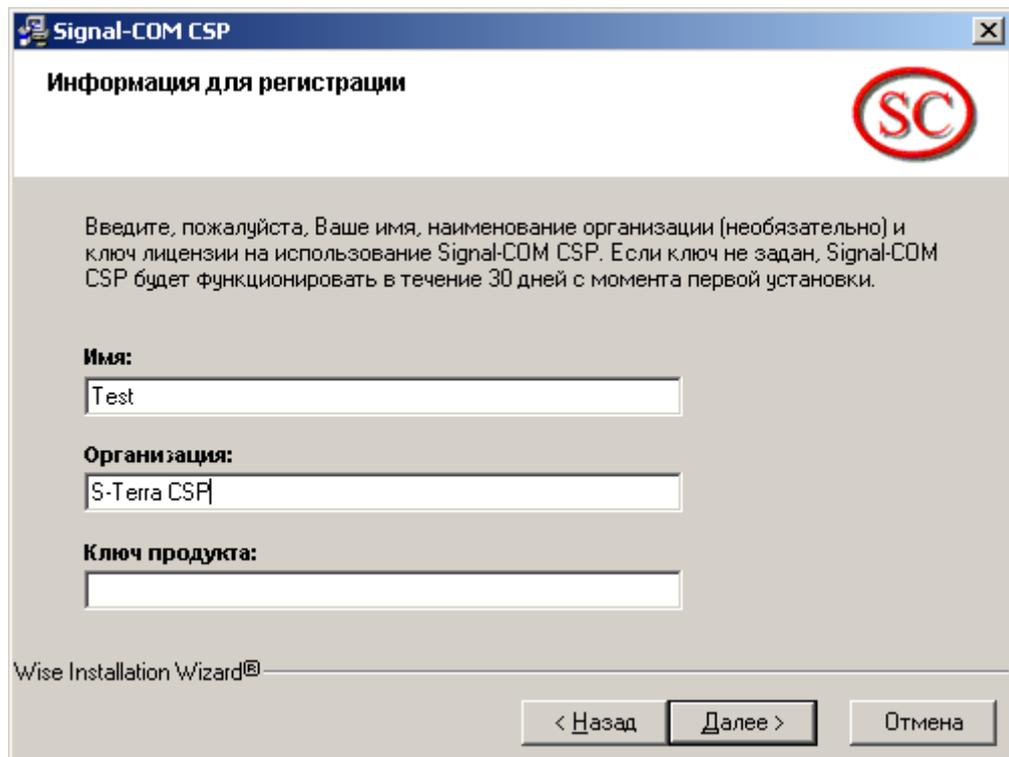


Рисунок 105

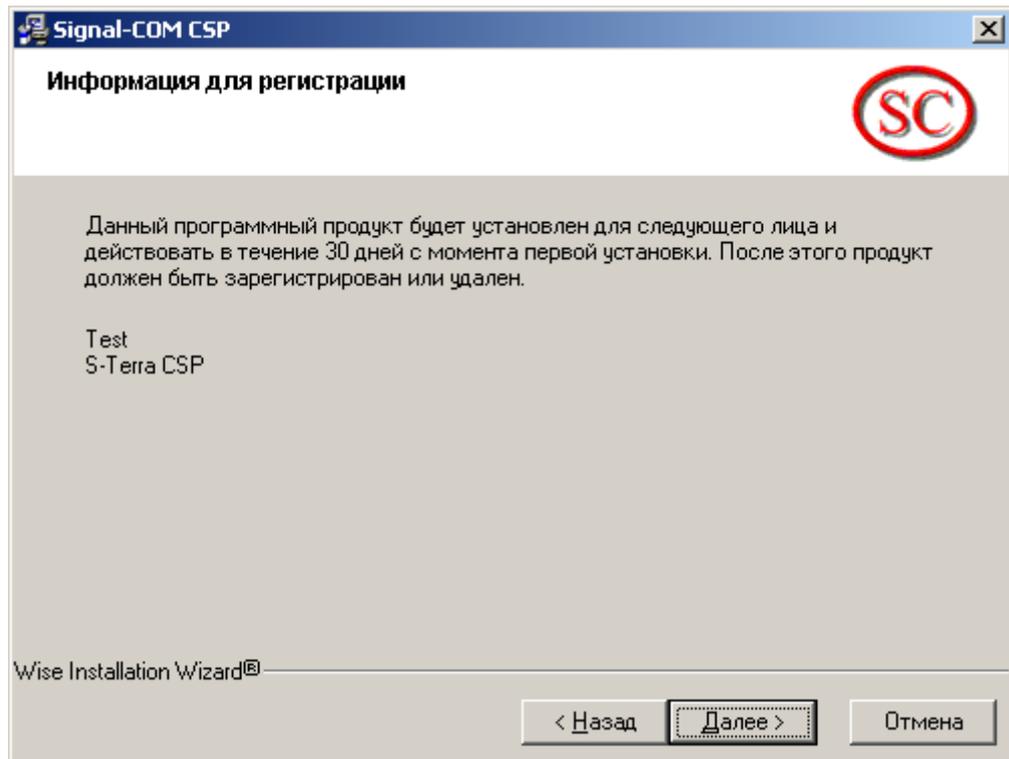


Рисунок 106

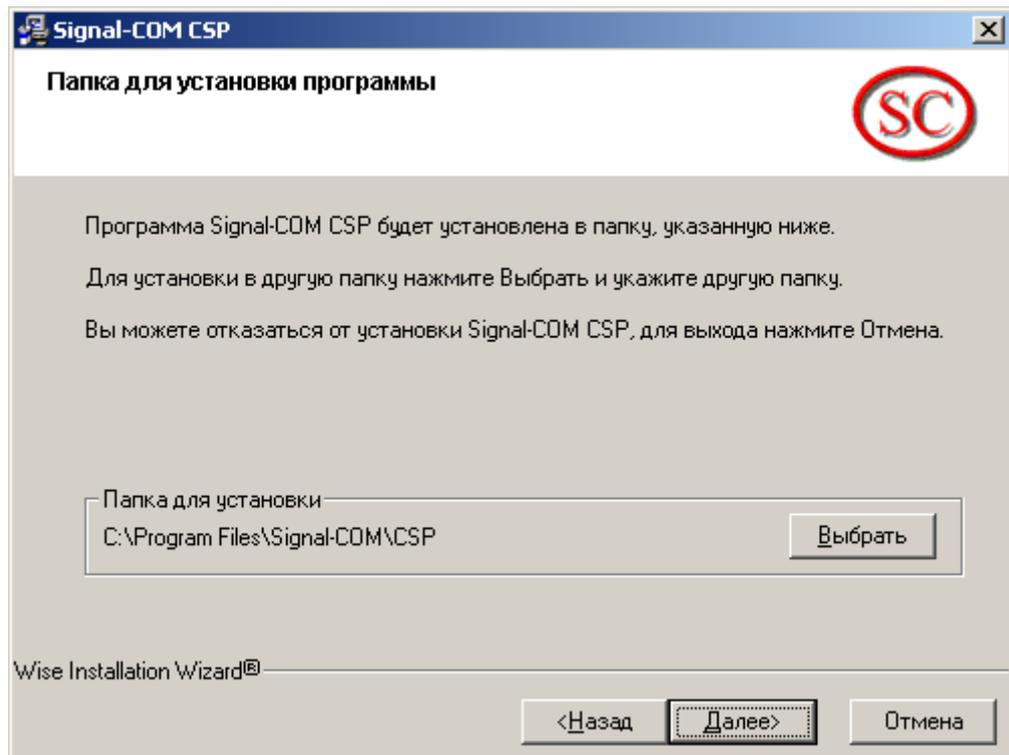


Рисунок 107

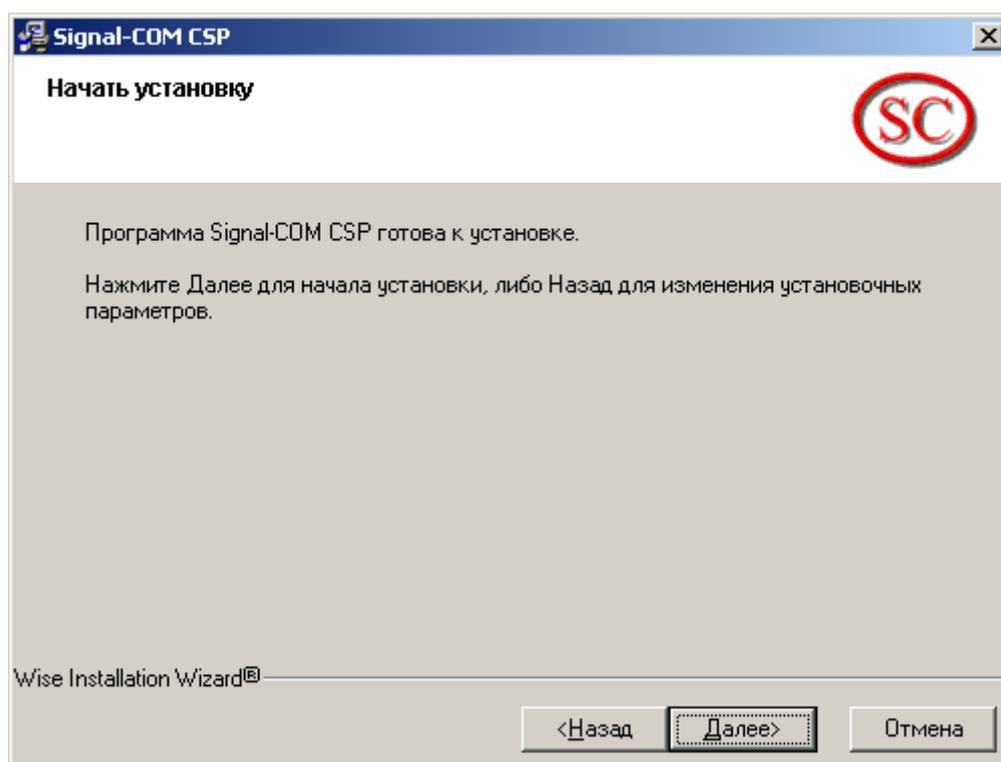


Рисунок 108

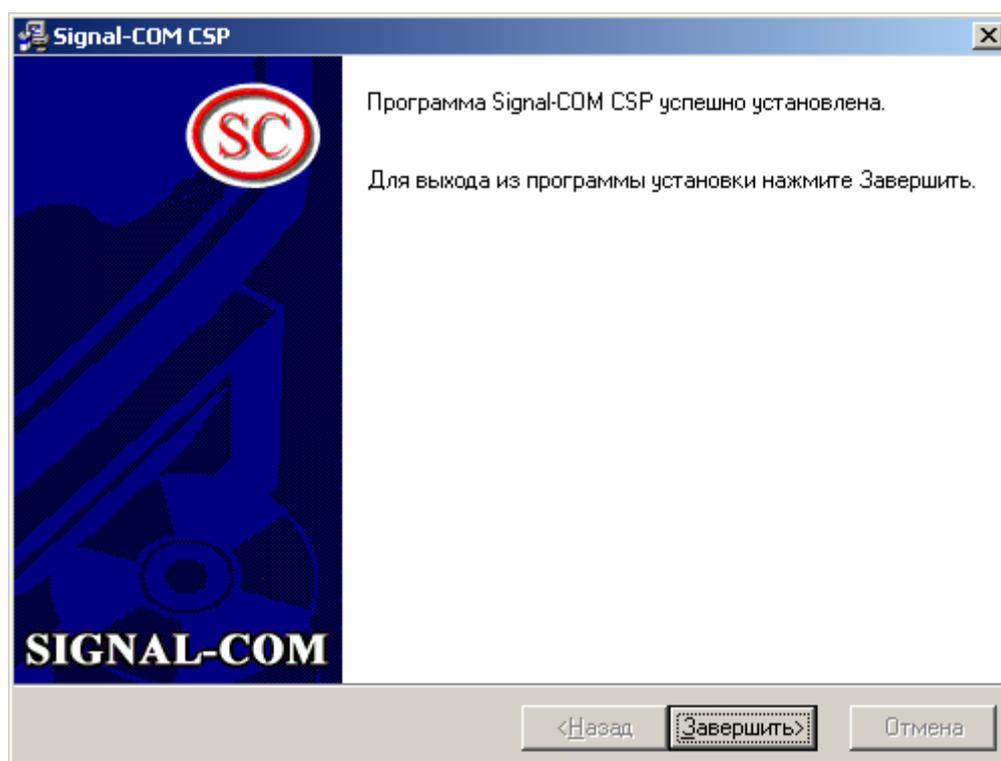


Рисунок 109

Инсталляция криптопровайдера завершена.

## 11.2. Установка и настройка Удостоверяющего Центра. Создание СА сертификата

На компьютере с установленной ОС Windows 2000 Server и включенным сервисом Internet Information Services (IIS) для инсталляции Удостоверяющего Центра Microsoft Certification Authority выполните следующее:

**Шаг1** : в окне установки компонент Windows (Start-Settings-Control Panel-Add/Remove Programs-Add/Remove Windows Components) установите флажок напротив Certificate Services и нажмите Next:

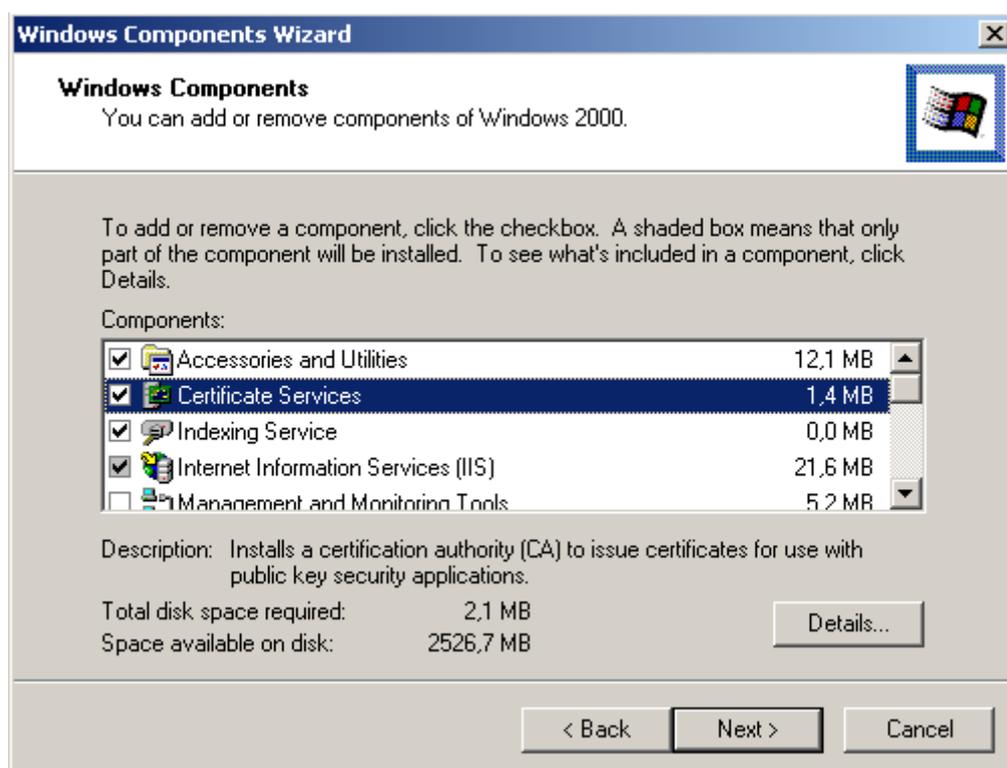


Рисунок 110

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

**Шаг2** : в следующем окне выберите Удостоверяющий центр с корневым СА сертификатом: поставьте переключатель в положение Stand-alone root CA. Установите флажок Advanced options и нажмите Next (Рисунок 111).

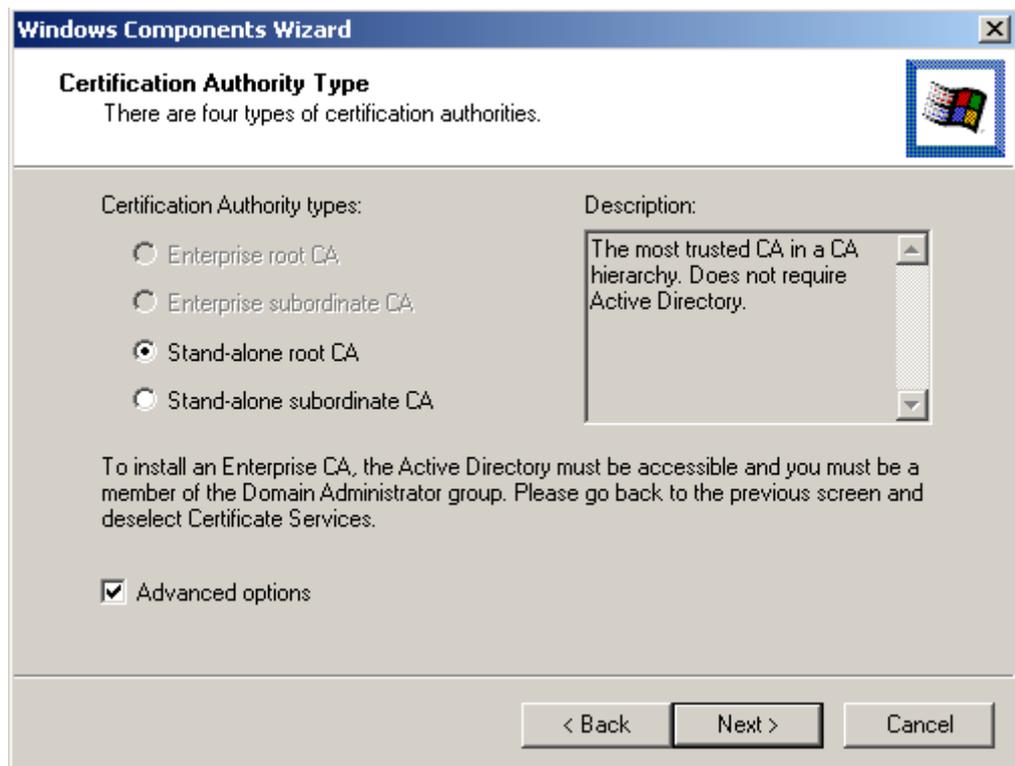


Рисунок 111

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

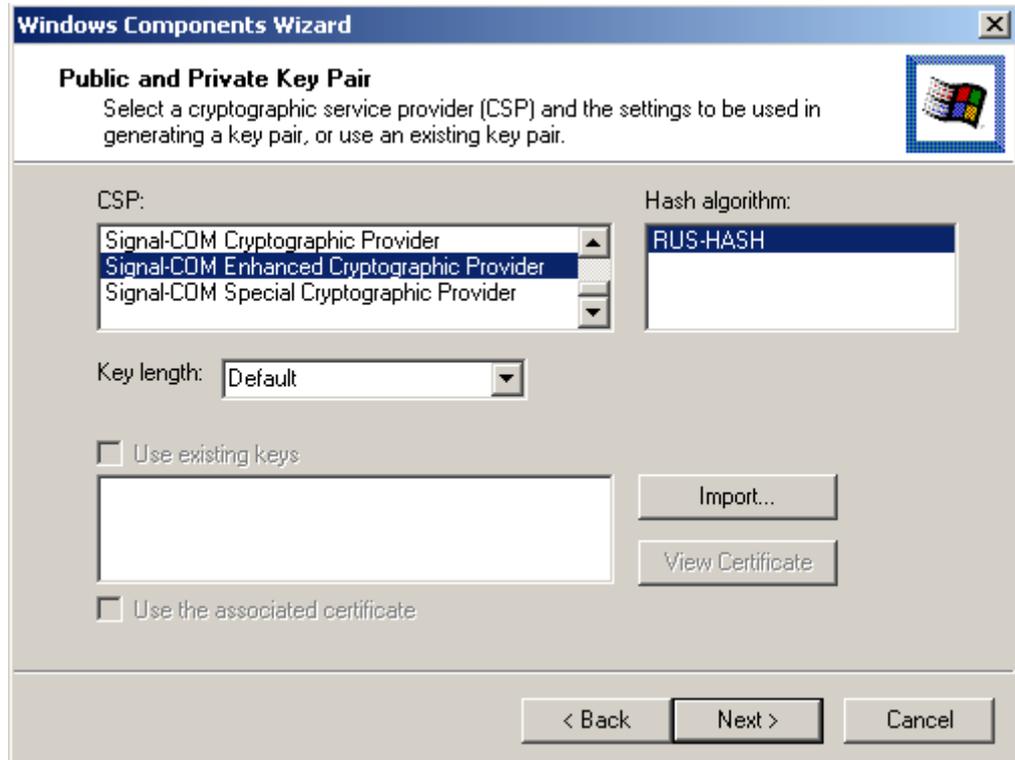
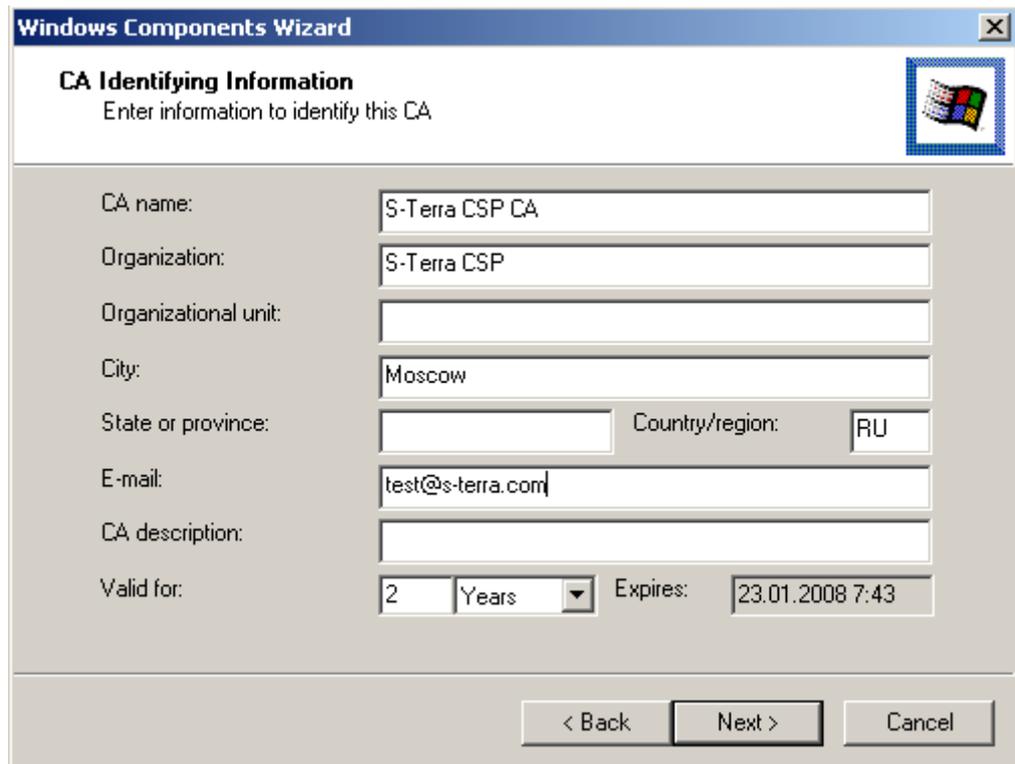


Рисунок 112

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



The screenshot shows the 'Windows Components Wizard' dialog box, specifically the 'CA Identifying Information' step. The title bar reads 'Windows Components Wizard' and the subtitle is 'CA Identifying Information'. Below the subtitle, it says 'Enter information to identify this CA'. The dialog contains several input fields: 'CA name' (S-Terra CSP CA), 'Organization' (S-Terra CSP), 'Organizational unit' (empty), 'City' (Moscow), 'State or province' (empty), 'Country/region' (RU), 'E-mail' (test@s-terra.com), 'CA description' (empty), and 'Valid for' (2 Years, Expires: 23.01.2008 7:43). At the bottom, there are three buttons: '< Back', 'Next >', and 'Cancel'.

Рисунок 113

**Шаг 5** : выберите ключевой носитель - Реестр, в котором будет записан контейнер с секретным ключом для CA сертификата и нажмите ОК :

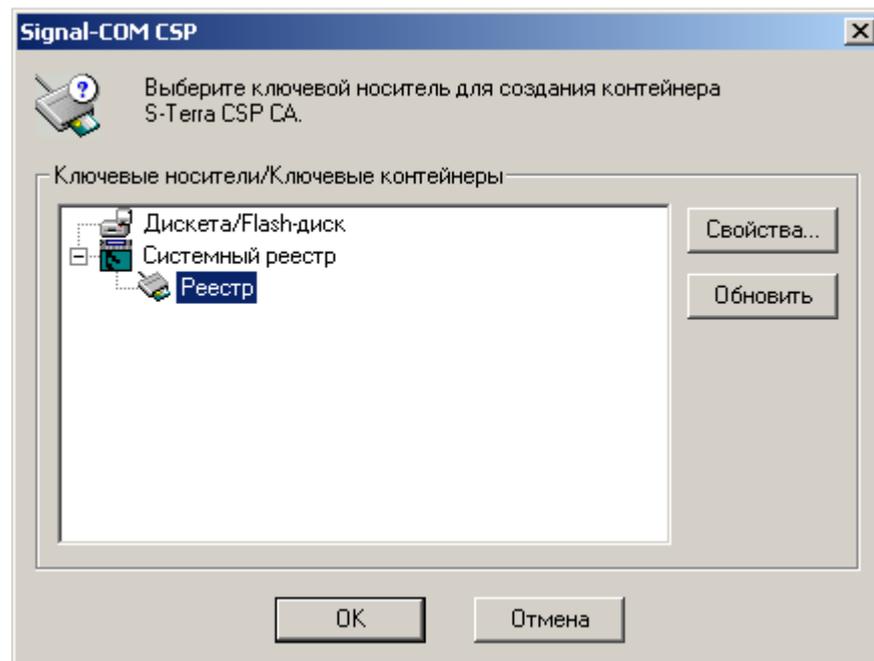


Рисунок 114

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

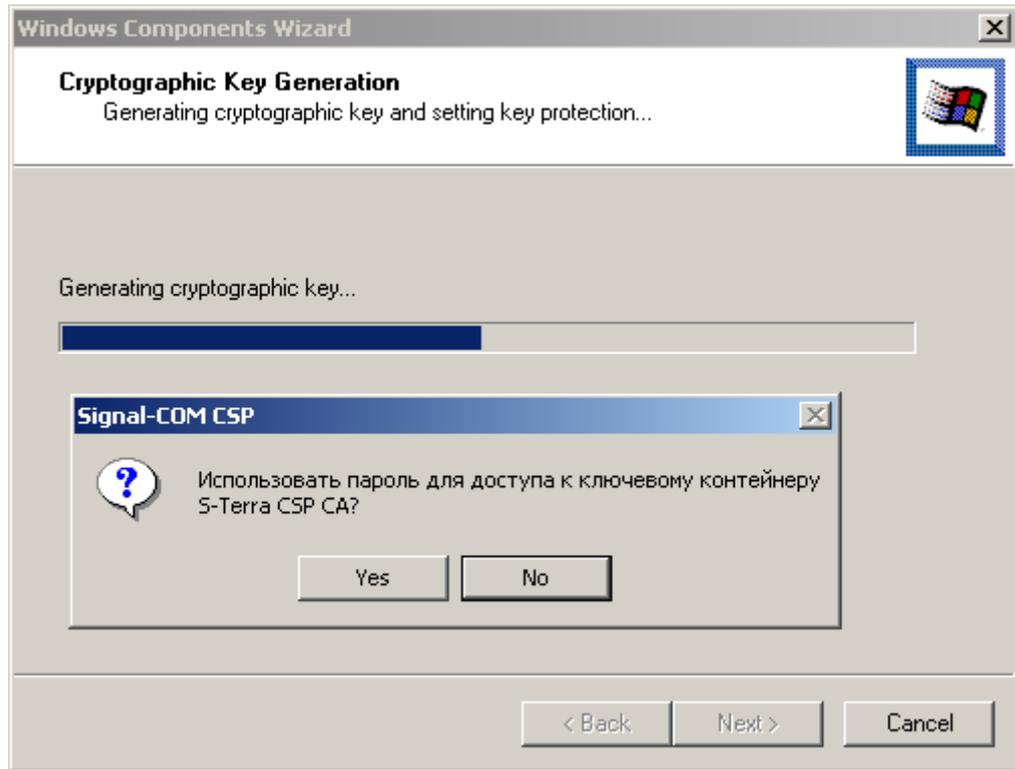


Рисунок 115

**Шаг 7:** задайте пароль к ключевому контейнеру и нажмите ОК:

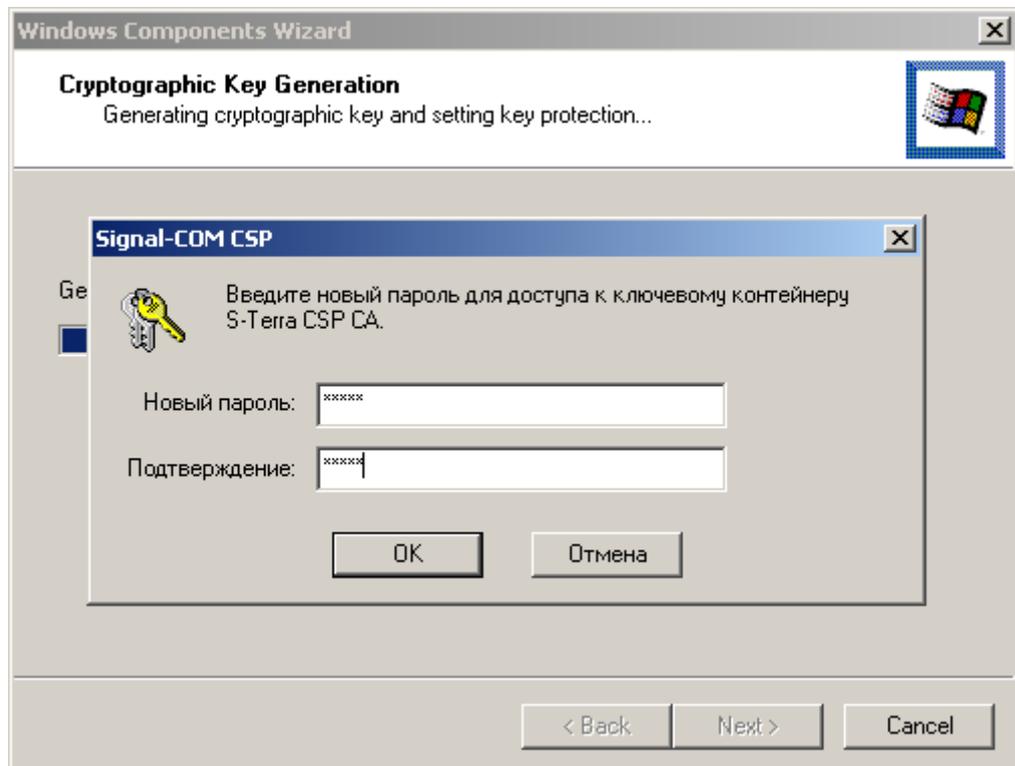


Рисунок 116

Шаг 8: для генератора случайных чисел вводите запрашиваемые символы:

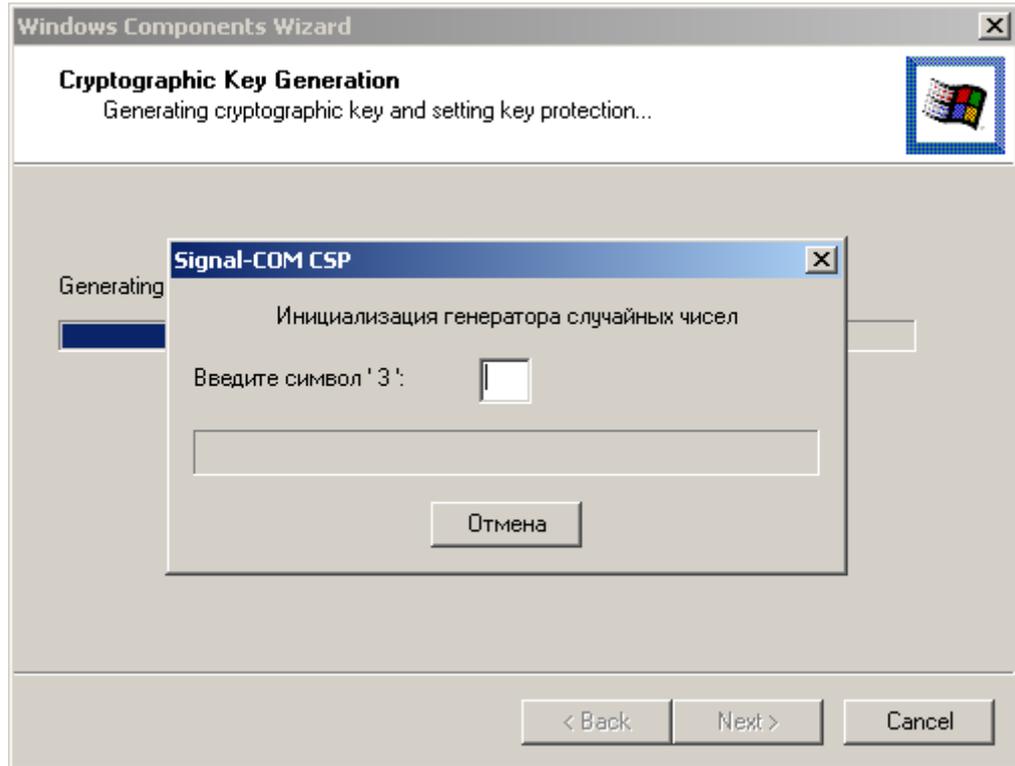


Рисунок 117

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

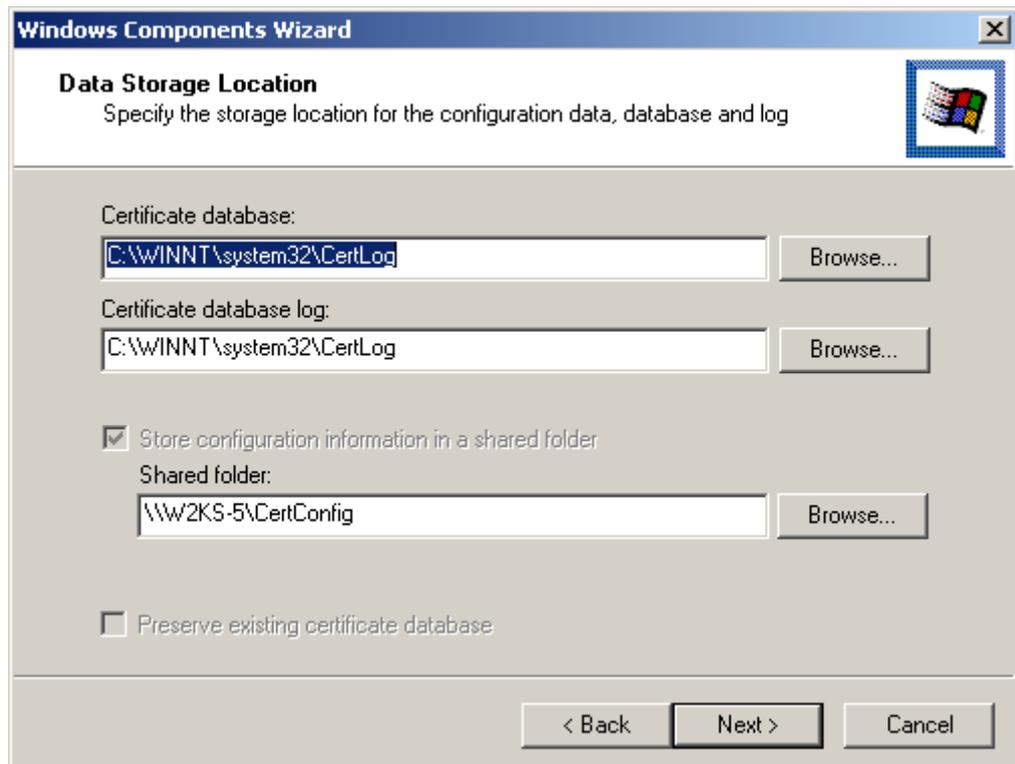


Рисунок 118

Шаг10: для остановки Internet Information Services нажмите OK:

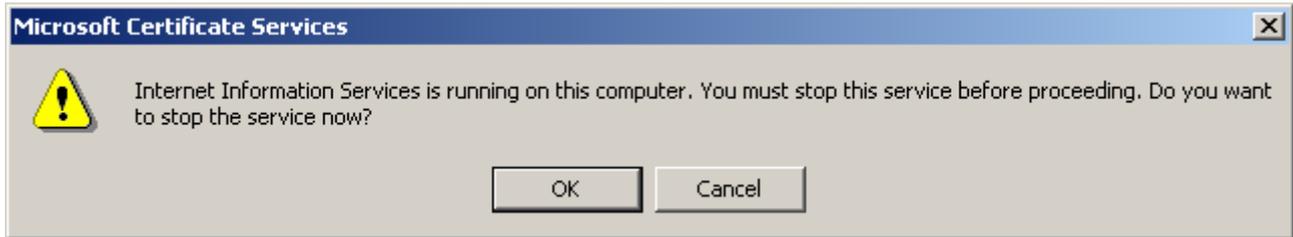


Рисунок 119

Шаг11: введите еще раз пароль к контейнеру с секретным ключом CA сертификата и нажмите OK:



Рисунок 120

Шаг12: инсталляция Удостоверяющего Центра завершена, нажмите Finish.



Рисунок 121

**Шаг13:** для автоматического создания подписываемых сертификатов войдите сначала в Certificate Authority (Start-Settings-Control Panel-Administrative Tools-Certificate Authority), а затем в Properties:

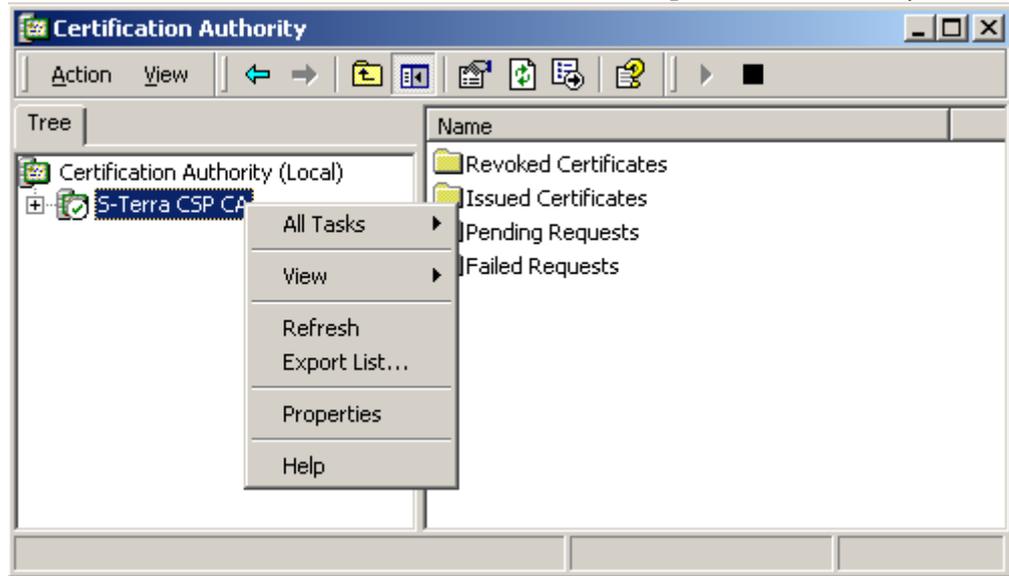


Рисунок 122

**Шаг14:** далее во вкладке Policy Module нажмите кнопку Configure...

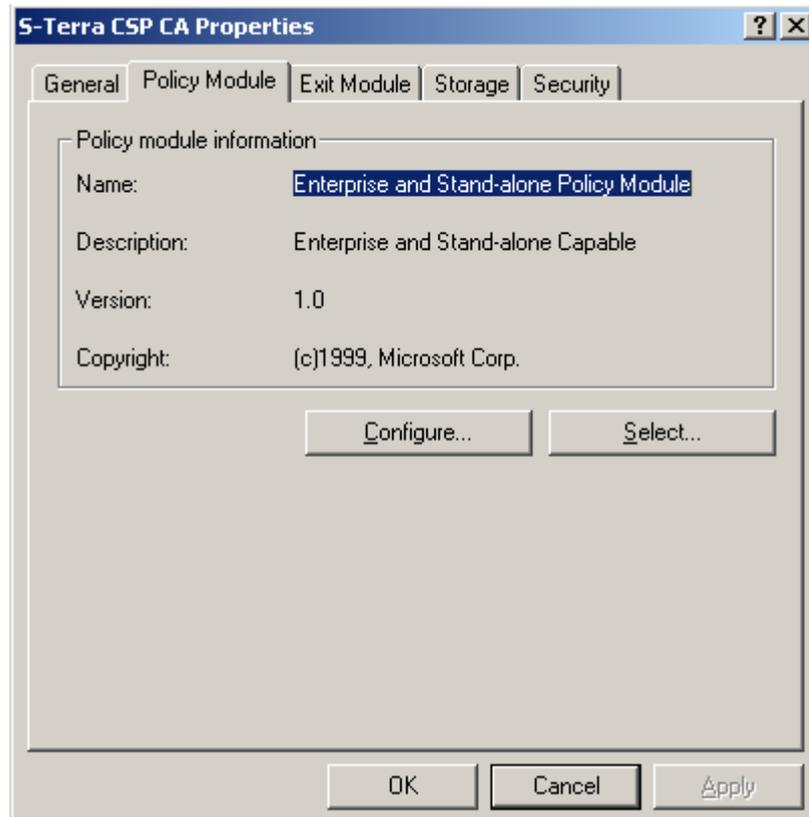


Рисунок 123

**Шаг 15:** и наконец, установите переключатель в положение Always issue the certificate (всегда издавать сертификат) во вкладке Default Action и нажмните OK:

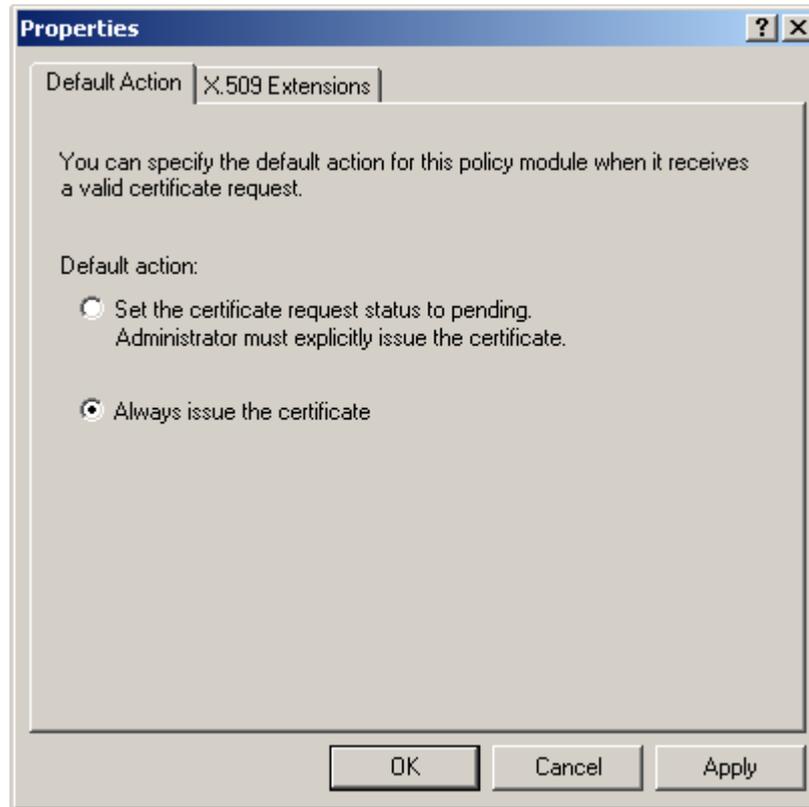


Рисунок 124

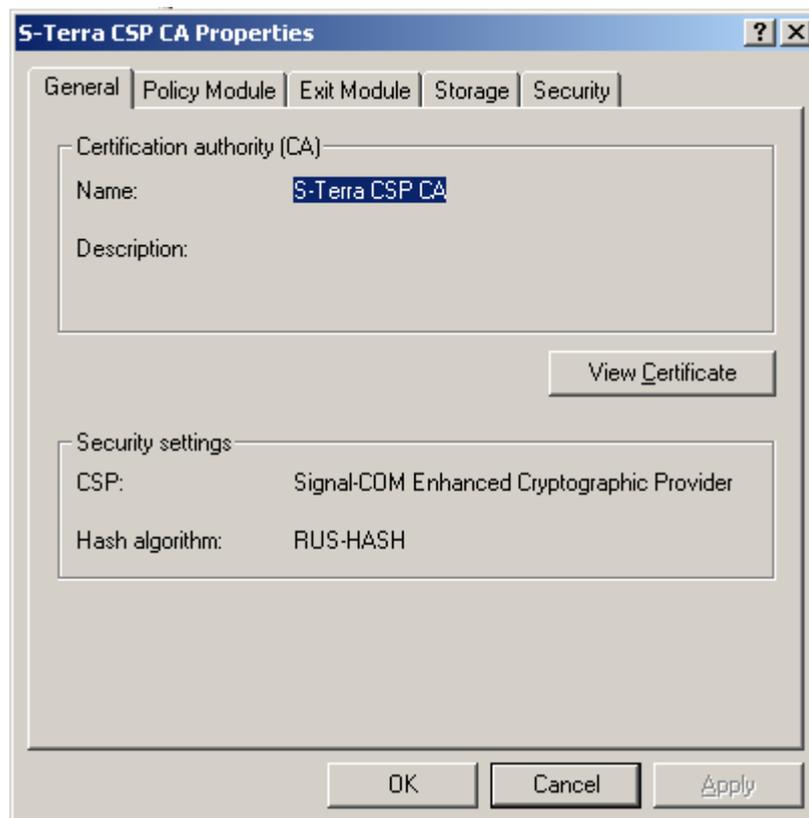


Рисунок 125

**Шаг16:** для просмотра CA сертификата во вкладке General нажмите кнопку View Certificate (Рисунок 125).

**Шаг17:** для копирования CA сертификата в файл, во вкладке Details нажмите кнопку Copy to File:

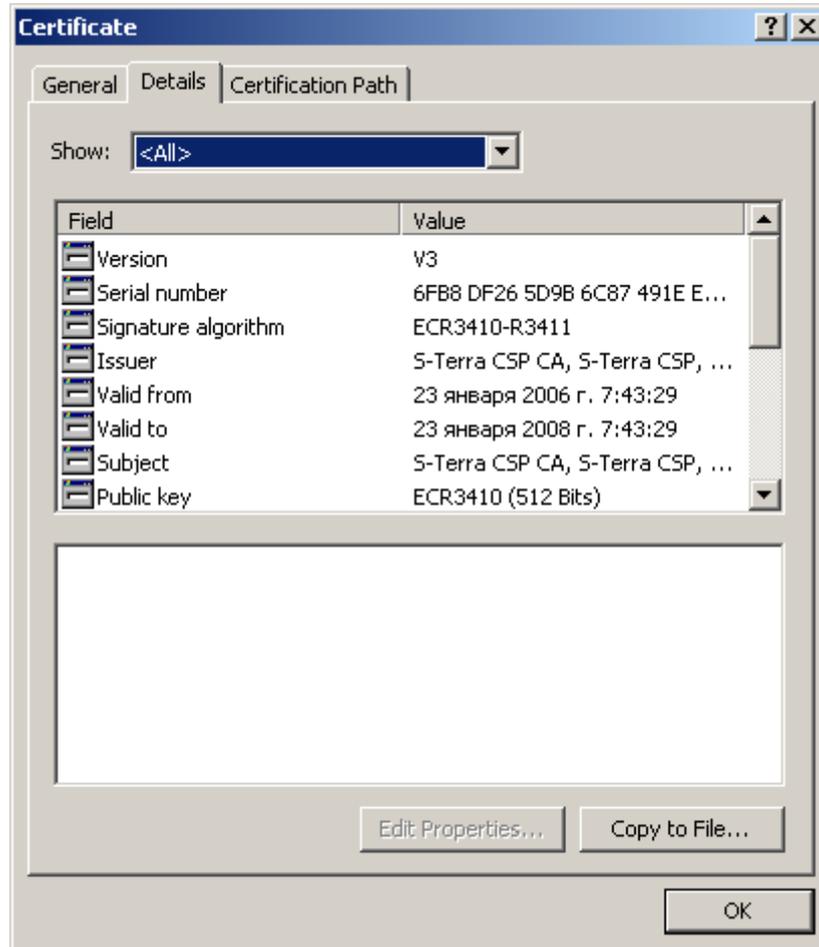


Рисунок 126

**Шаг18:** далее в открываемых окнах визарда Certificate Export введите имя файла для копирования CA сертификата, например на жесткий диск.

## 11.3. 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/](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:

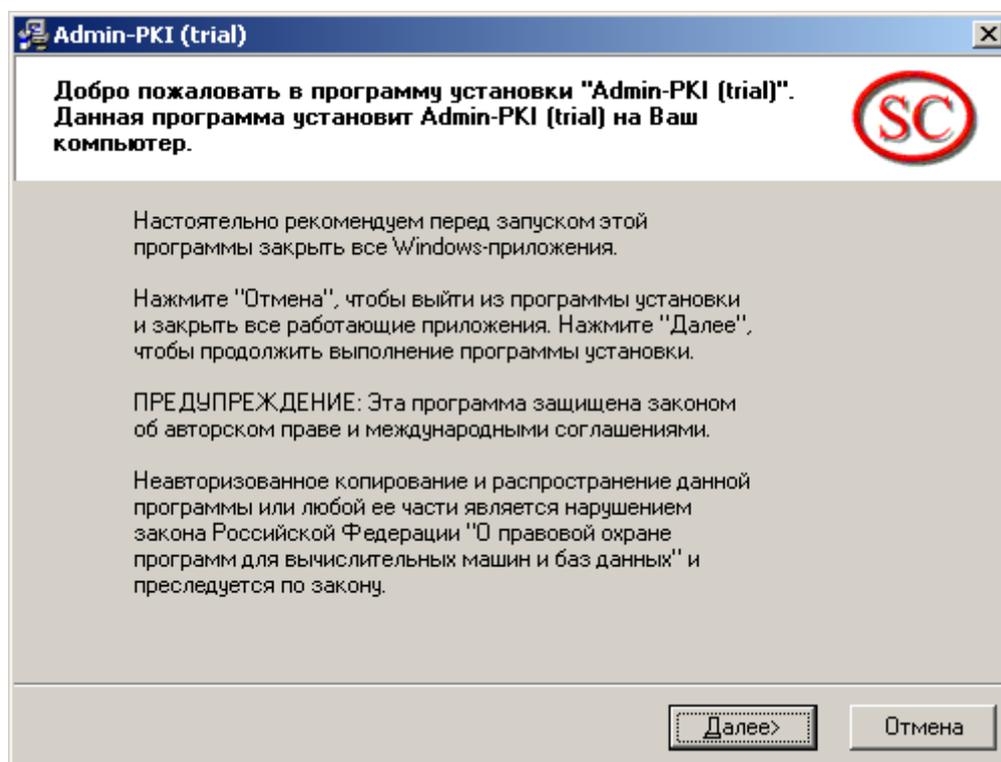


Рисунок 127

Далее везде нажимайте кнопку Далее:

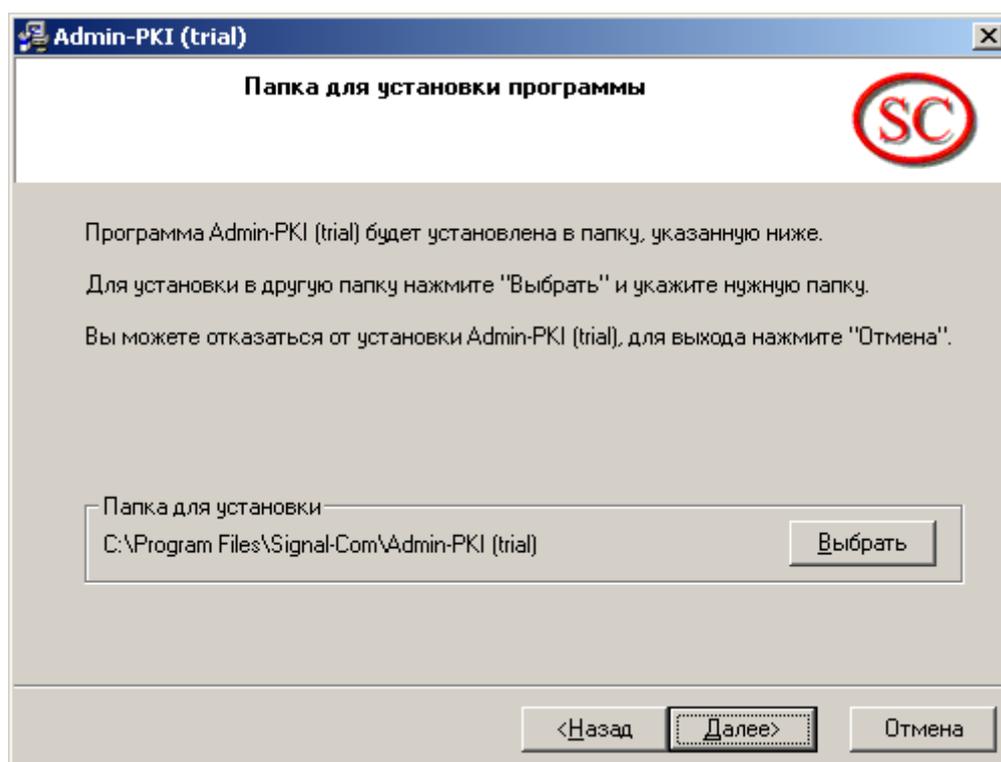


Рисунок 128

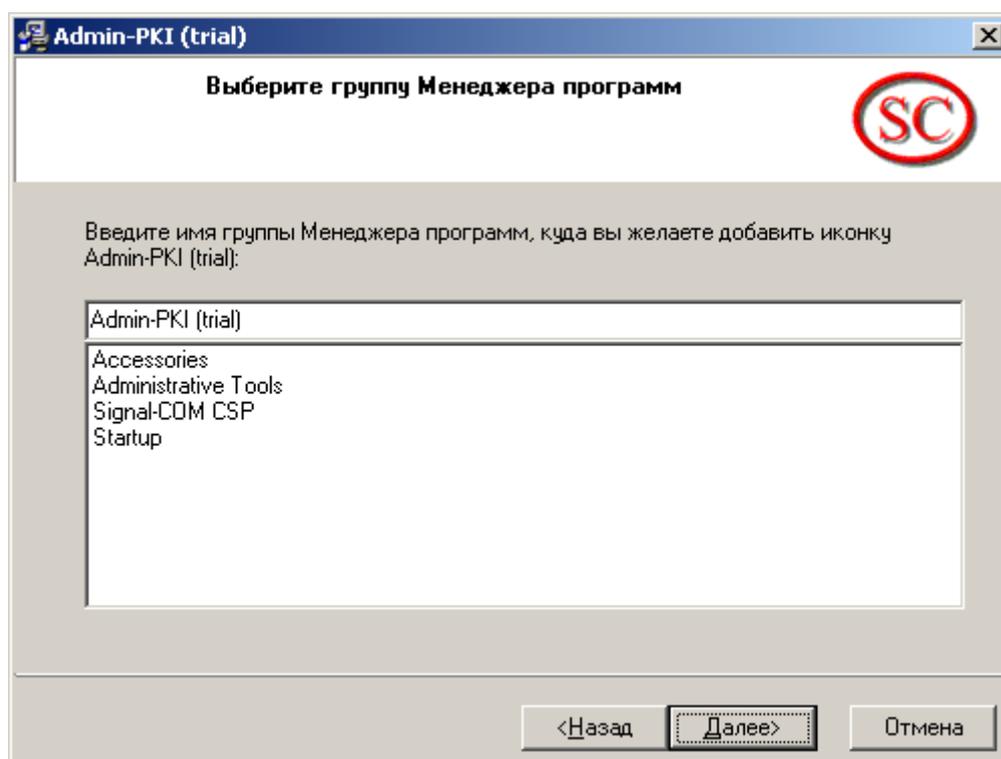


Рисунок 129

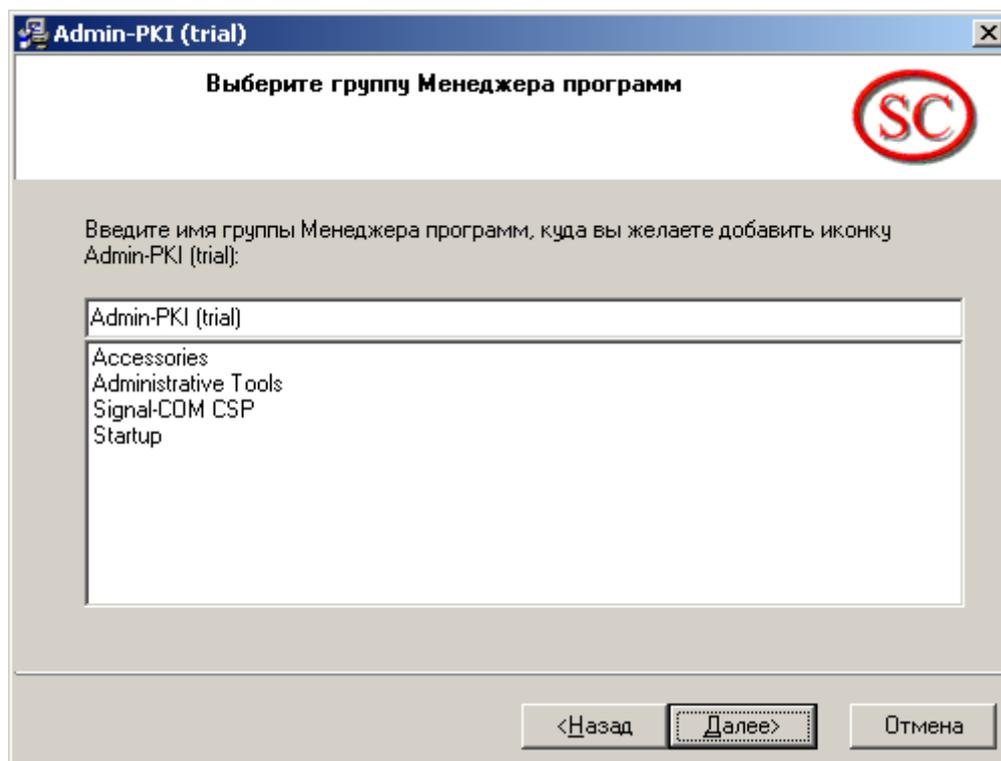


Рисунок 130

Инсталляция завершена, нажмите Завершить.

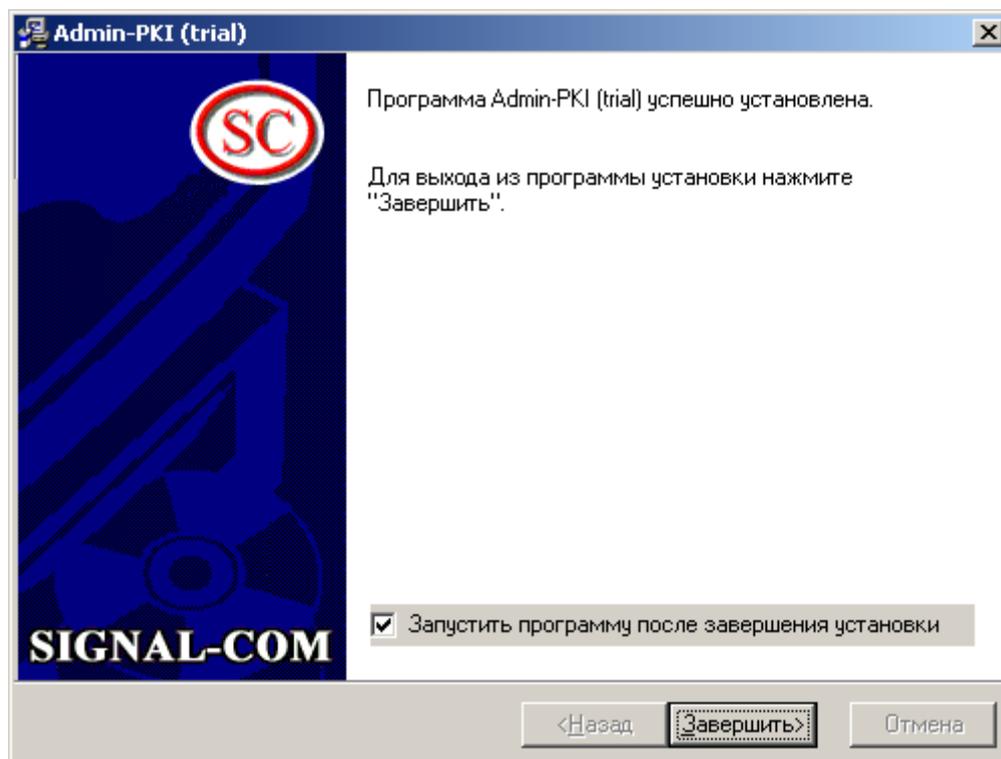


Рисунок 131

## 11.4. 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 СА 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:

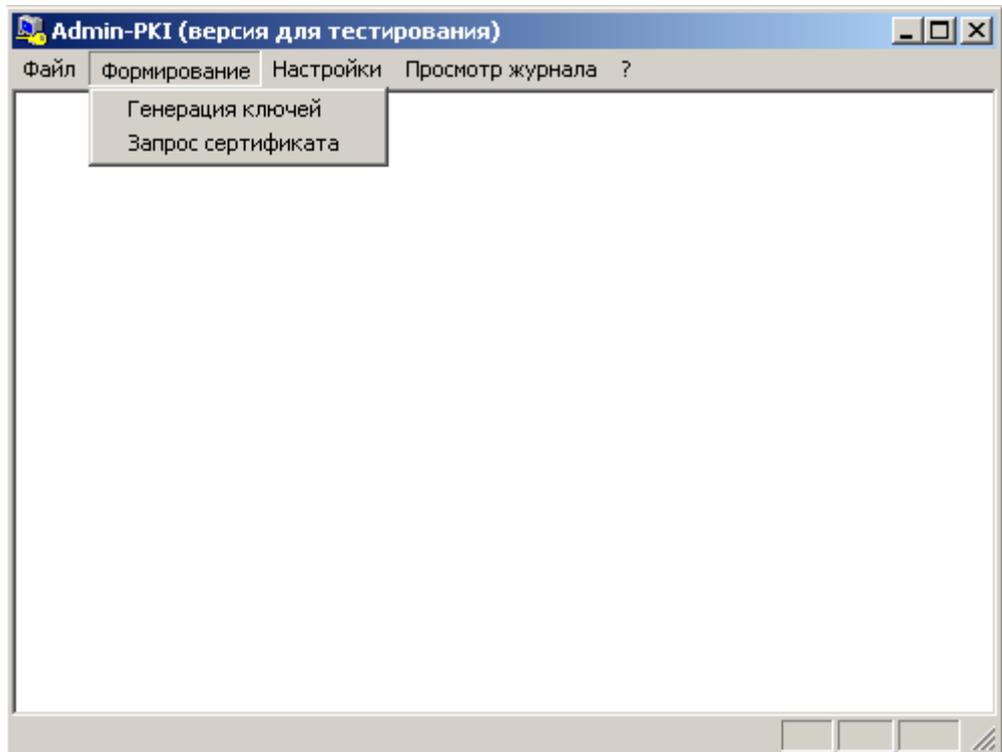


Рисунок 132

**Шаг2:** в поле Каталог ключевого носителя задайте контейнер в виде каталога.

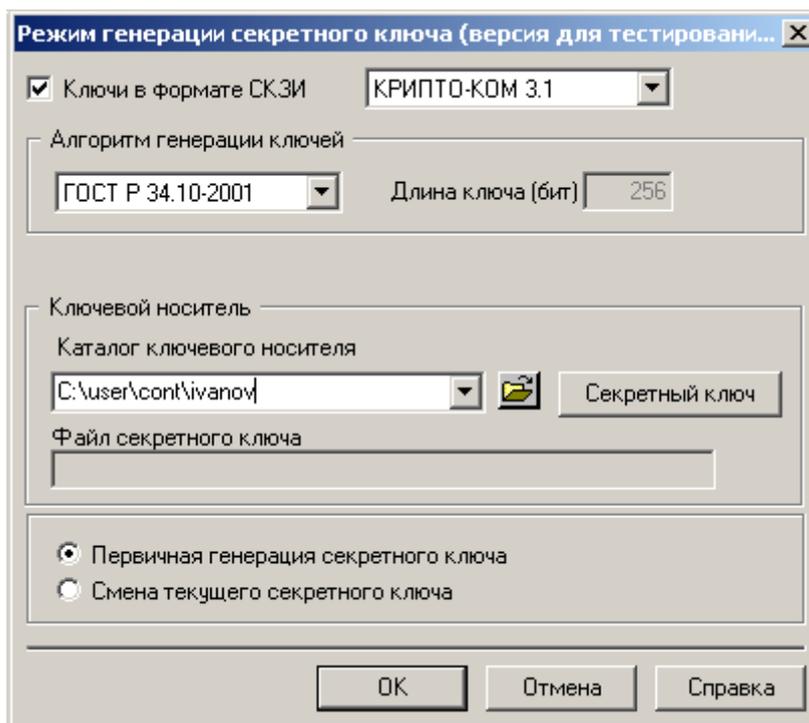


Рисунок 133

**Шаг3:** 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 ОК:

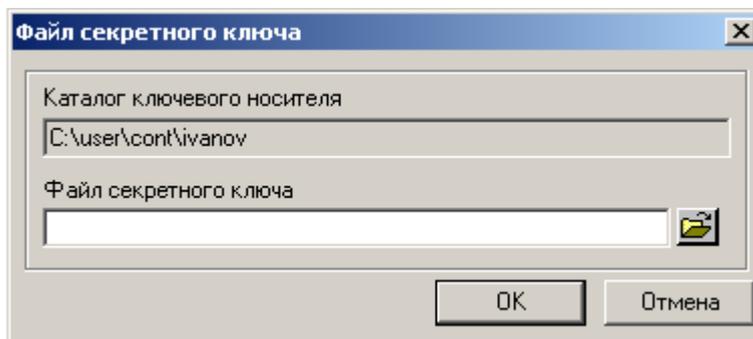


Рисунок 134

**Шаг4:** на вопрос о задании нового каталога (контейнера) ответьте Yes:

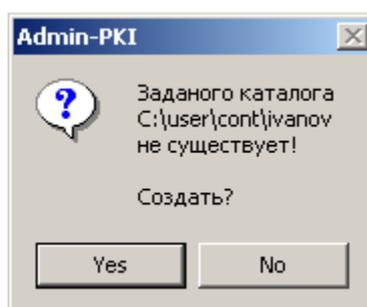


Рисунок 135

**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 ОК:

Параметры запроса сертификата (версия для тестирования) ? X

Ключи в формате СКЗИ

Каталог ключевого носителя с ключом для формирования запроса  
C:\user\cont\ivanov [Секретный ключ]

Файл запроса  
C:\user\cont\ivanov\request.pem

Тип запроса  
 Самоподписанный  
 Подписанный другим ключом

Каталог ключевого носителя с ключом для подписи запроса  
A:\ [Секретный ключ]

Сертификат ключа для подписи запроса

Кодировка символов в запросе  ANSI  UTF8

Запрашиваемые параметры сертификата

Страна RU (RU для России) [Поиск...]

Область/Район

Город/Село Moscow

Организация S-Terra

Подразделение devel

Должность

Полное имя Ivanov

E-mail адрес ivanov@s-terra.com

OK Отмена Справка

Рисунок 136

**Шаг 6:** в окне отчета о выполненных операциях нажмите ОК:

Admin-PKI X

Процедура генерации ключей СКЗИ и запроса успешно завершена !

Каталог ключевого носителя:  
C:\user\cont\ivanov

Файл запроса на сертификат:  
C:\user\cont\ivanov\request.pem

Файл запроса необходимо передать на сертификацию.  
Работа с новым секретным ключом будет возможна только после получения сертификата.

OK

Рисунок 137

**Шаг7:** в окне на вопрос о сохранении в буфере обмена запроса на сертификат нажмите Yes :

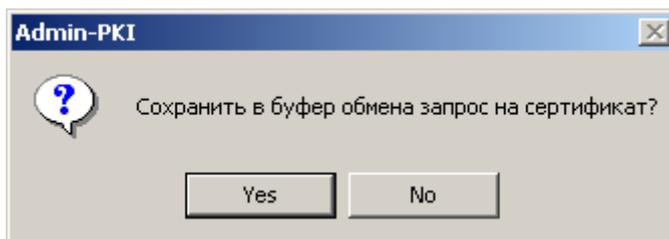


Рисунок 138

**Шаг8:** на вопрос о передаче запроса на сертификат в Удостоверяющий Центр по E-mail нажмите No :

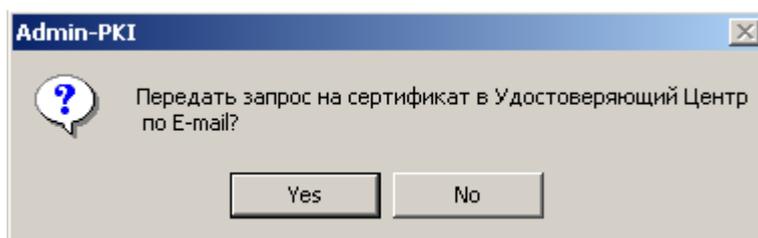


Рисунок 139

**Шаг9:** на вопрос о передаче запроса на сертификат в Удостоверяющий Центр через Web-интерфейс Удостоверяющего Центра нажмите No :

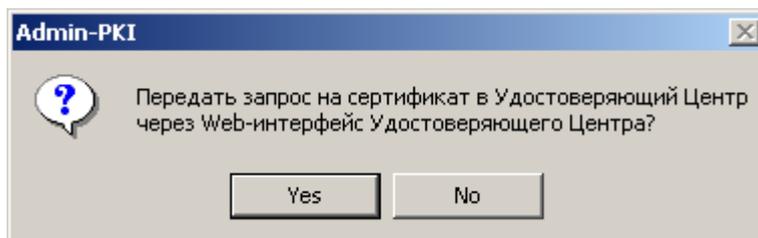


Рисунок 140

**Шаг10:** на вопрос о распечатке запроса на сертификат нажмите No :

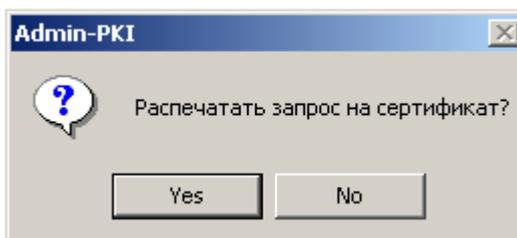


Рисунок 141

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.

## 11.5. 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.

**Шаг1:** 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://10.0.32.4/certsrv`.

**Шаг2:** в появившемся окне Удостоверяющего Центра выберите задачу – установите переключатель в положение `Request a certificate` и нажмите `Next`:

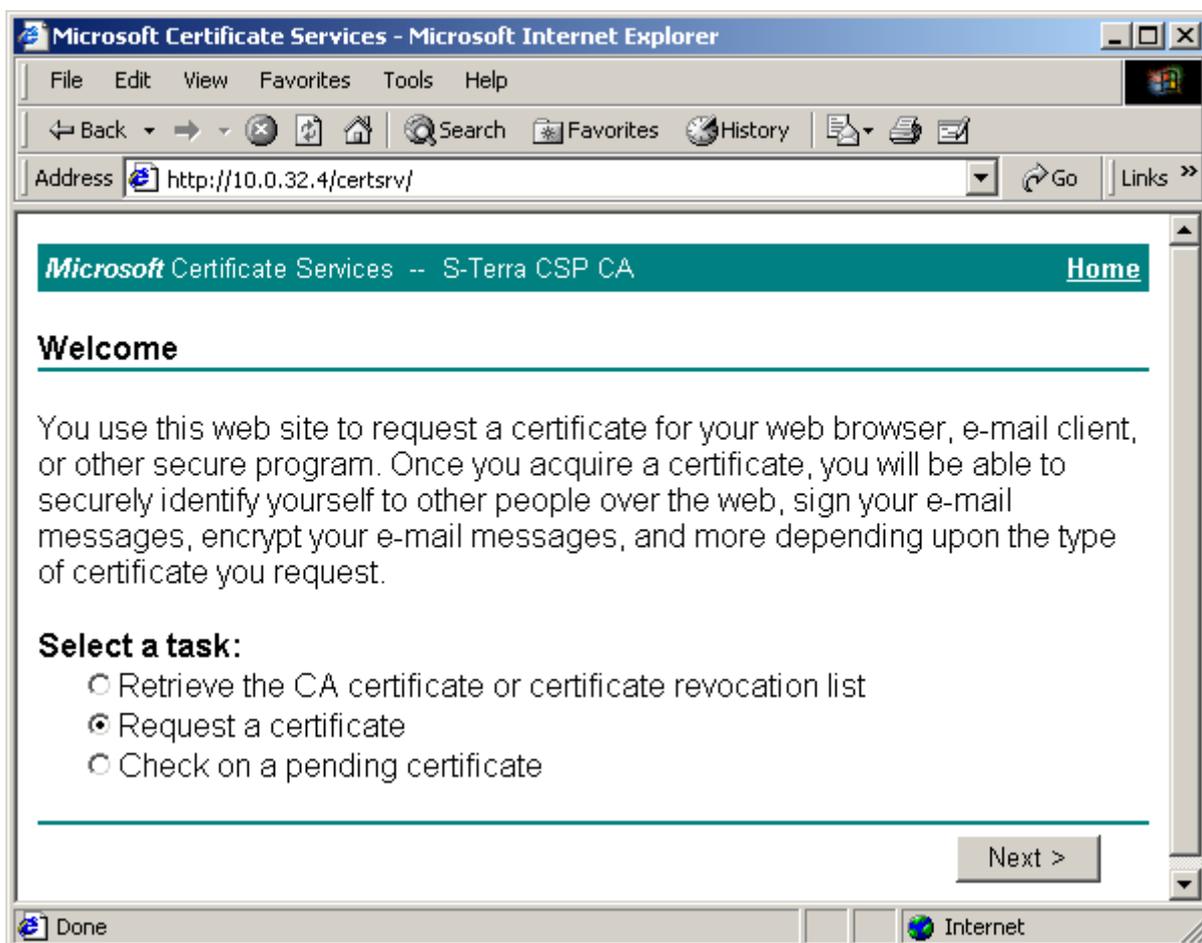


Рисунок 142

**Шаг 3:** поставьте переключатель в положение *Advanced request* и нажмите *Next* (Рисунок 143) :

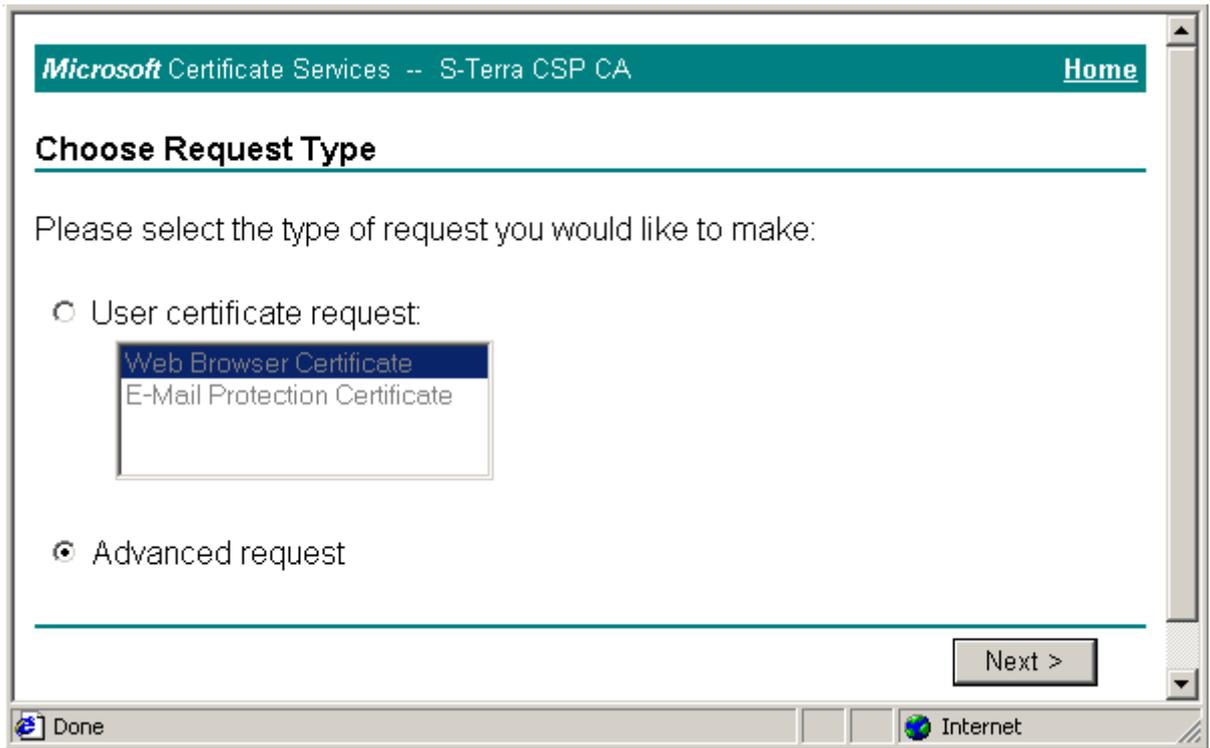


Рисунок 143

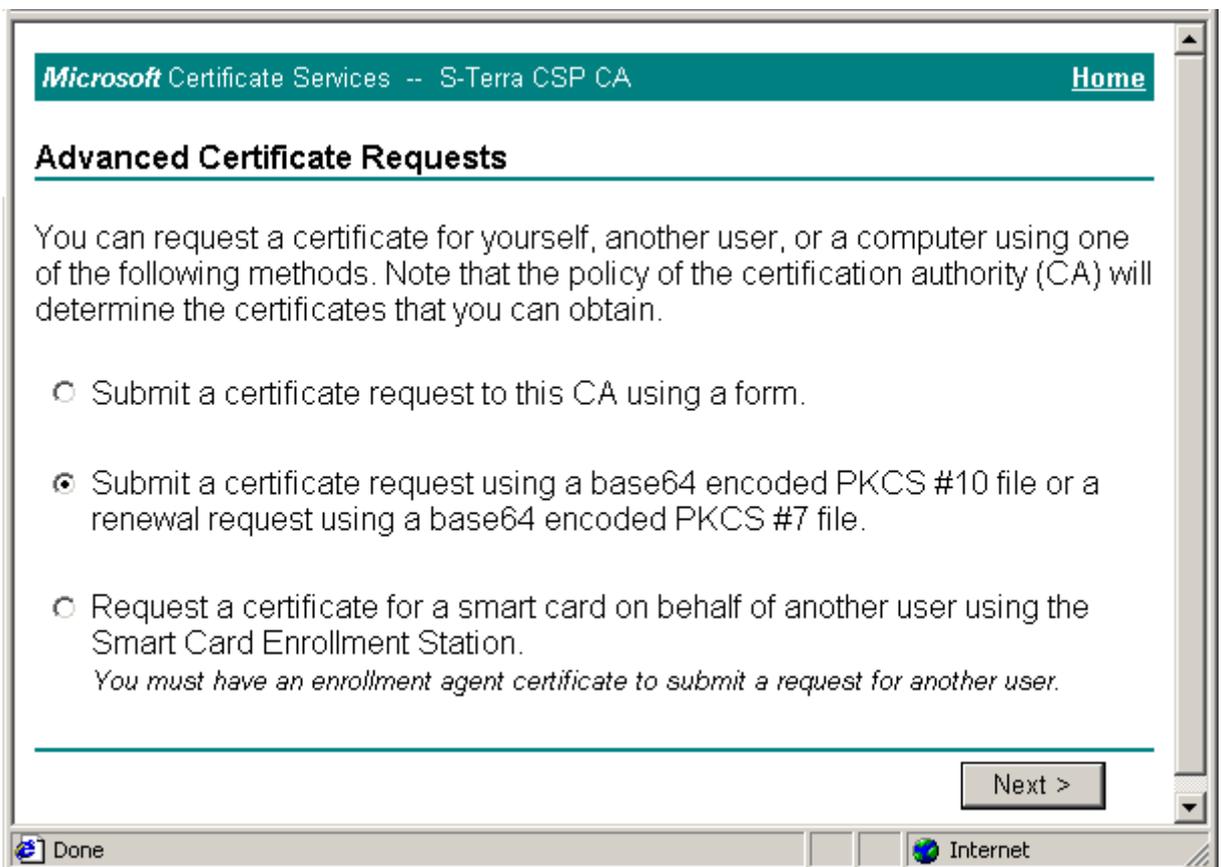


Рисунок 144

**Шаг4** : запрос на сертификат пользователя записан в файл в формате PKCS#10, поэтому для указания вида в каком будет отослан запрос на сертификат, поставьте переключатель во второе положение и нажмите Next (Рисунок 144) .

**Шаг5** : 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 :

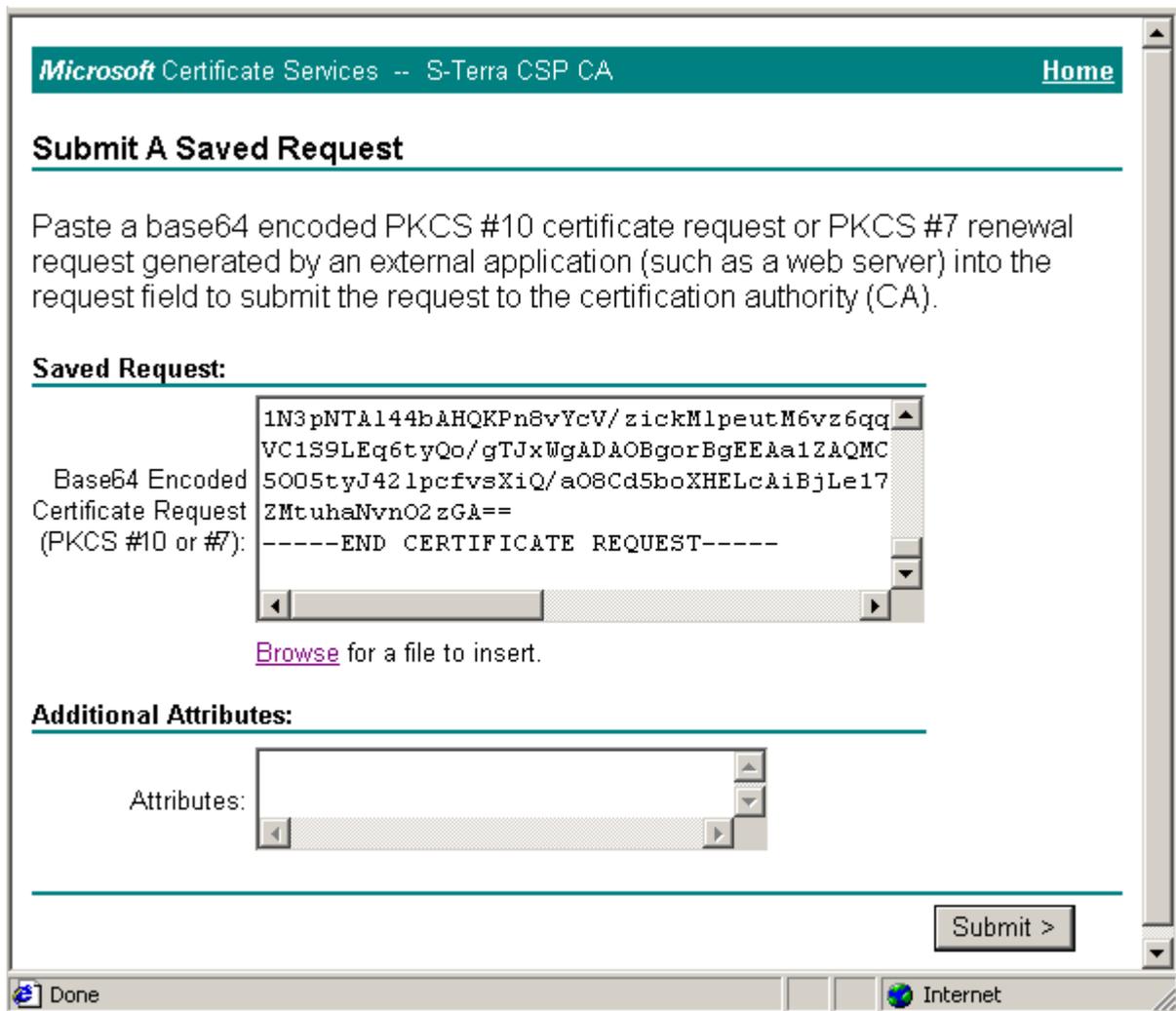


Рисунок 145

**Шаг6** : Удостоверяющий Центр издал локальный сертификат по полученному запросу. Для загрузки сертификата в файл нажмите Download CA certificate(!!!) (Рисунок 146).

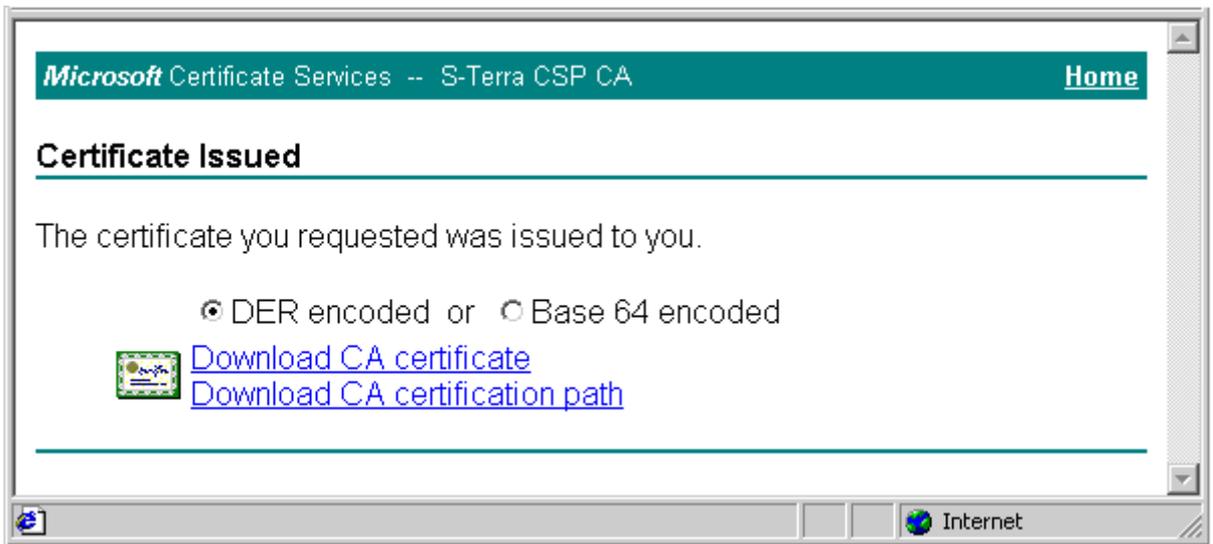


Рисунок 146

**Шаг 7:** установите переключатель в положение *Save this file to disk* и нажмите **OK**:



Рисунок 147

**Шаг 8:** введите имя файла, в который будет записан локальный сертификат и нажмите **Save** (Рисунок 148) .

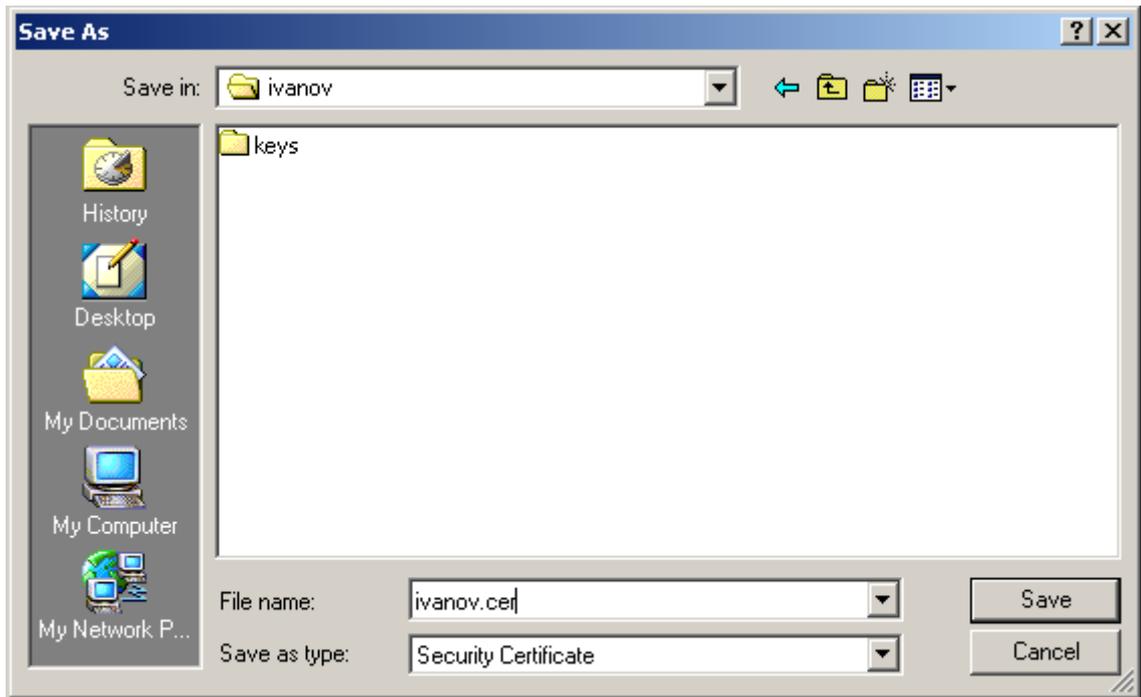


Рисунок 148

В результате всех выполненных действий мы создали CA сертификат и локальный сертификат для шлюза безопасности, 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. Сертификаты и контейнер доставьте на шлюз безопасности любым доступным способом.

## 11.6. 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 сертификат. Локальный сертификат нужно доставить любым способом на аппаратно-программный комплекс и утилитой `cert_mgr import` зарегистрировать сертификат в базе Продукта.

## 12. Создание локального сертификата с использованием СКЗИ "LirSSL"

Настройка шлюзов безопасности CSP VPN Gate и CSP RVPN производится одинаково, выполняют они одни и те же функции, поэтому в дальнейшем будем использовать только одно наименование - CSP VPN Gate или Продукт.

Создание локального сертификата для шлюза безопасности можно осуществить по одному из двух вариантов, которые описаны в этой главе. В обоих вариантах используются утилиты, созданные компанией "ЛИССИ", и которые входят в состав дистрибутива Продукта CSP VPN Gate. Утилиты используются для создания ключевой пары, запроса на локальный сертификат, создания контейнера PKCS#12, проверки целостности этого контейнера и др.

Список этих утилит следующий - `make_key_pair`, `make_pkcs10`, `pkcs10_info`, `make_pkcs12`, `checkcont`, `getcert`. После инсталляции Продукта все утилиты размещены в каталоге `/opt/lissi/bin`.

### Вариант 1

**Шаг1:** все действия по созданию ключевой пары, формированию запроса и созданию локального сертификата для шлюза безопасности выполняются администратором СА на рабочем месте с установленным программным комплексом "ЛИССИ-УЦ/LISSICA".

**Шаг2:** получите от администратора СА по заслуживающему доверия каналу связи СА сертификат Удостоверяющего Центра, локальный сертификат и секретный ключ локального сертификата в ключевом контейнере PKCS#12.

**Шаг3:** доставьте на шлюз с инсталлированным Продуктом CSP VPN Gate любым доступным способом СА сертификат в виде файла, ключевой контейнер PKCS#12 с локальным сертификатом и секретным ключом.

**Шаг4:** проверьте целостность доставленного контейнера путем вычисления специального дайджеста с вводом пароля и сравнения его с дайджестом, хранящимся в контейнере. Для этого используйте утилиту `checkcont`:

```
cspvpn:/opt/lissi/bin# ./checkcont container_file_name
password
```

где

<code>container_file_name</code>	имя файла с ключевым контейнером PKCS#12
<code>password</code>	пароль к контейнеру

Если проверка завершилась успешно, то утилита выдает в стандартный вывод сообщение "Keystore digest is OK".

Пример команды:

```
cspvpn:/opt/lissi/bin# ./checkcont gate1_cont12.p12 12345
```

**Шаг5:** экспортируйте локальный сертификат из контейнера PKCS#12 в отдельный файл при помощи утилиты `getcert`:

```
cspvpn:/opt/lissi/bin# ./getcert container_file_name password
[alias] certificate_file_name
```

где

<code>container_file_name</code>	имя файла с ключевым контейнером PKCS#12
<code>password</code>	пароль к контейнеру
<code>alias</code>	алиас (имя ключевой пары), по которому будет происходить поиск ключевой пары в контейнере. При его отсутствии будет рассматриваться первая попавшаяся ключевая пара
<code>certificate_file_name</code>	имя файла, в который будет экспортирован найденный локальный сертификат из контейнера.

При этом секретный ключ локального сертификата не расшифровывается и не извлекается из контейнера.

Пример команды:

```
cspvpn:/opt/lissi/bin# ./getcert gate1_cont12.p12 12345 gate1
loc_cer.crt
```

**Шаг 6:** зарегистрируйте СА и локальные сертификаты в Продукте CSP VPN Gate, используя утилиту `cert_mgr` из состава Продукта. Такая регистрация описана в документе [“Специализированные команды”](#).

## Вариант 2

Создание ключевой пары и формирование запроса на локальный сертификат шлюза безопасности выполняются администратором безопасности на программно-аппаратном комплексе CSP VPN Gate или модуле NME-RVPN. Опишем все эти действия подробно.

**Шаг 1:** установите Продукт CSP VPN Gate на программно-аппаратный комплекс или на модуль NME-RVPN. Установка описана в документе [“Установка CSP VPN Gate при использовании СКЗИ “LirSSL”](#) или [“Установка CSP VPN Gate на модуль”](#). Во время установки создается файл `prng_start.bin`, в который записывается начальное значение датчика случайных чисел.

**Шаг 2:** создайте ключевую пару локального сертификата, используя утилиту `make_key_pair`, созданную компанией “ЛИССИ”. Выполните команду:

```
cspvpn:/opt/lissi/bin# ./make_key_pair -out <file>
[-gostsign_param <gost3410_param>] [-gosthash_param
<gost3411_param>]
```

где

<code>-out &lt;file&gt;</code>	имя файла, в который записывается ключевая пара, формат файла – PEM
<code>-gostsign_param &lt;gost3410_param&gt;</code>	набор параметров КриптоПро для ГОСТ Р 34.10-2001. Рекомендуемое значение и значение по умолчанию - <code>par_ecc_a</code> .
<code>-gosthash_param &lt;gost3411_param&gt;</code>	набор параметров КриптоПро для ГОСТ Р 34.11-94. Рекомендуемое значение и значение по умолчанию <code>par_hash_1</code> .

При создании ключей используются криптографические алгоритмы ГОСТ Р 34.10-2001 для ЭЦП и ГОСТ Р 34.11-94 для функции хэширования.

Пример команды:

```
cspvpn:/opt/lissi/bin# ./make_key_pair -out pair_key
```

**Шаг3** : создайте запрос на локальный сертификат, используя утилиту `make_pkcs10`. Для этого выполните команду:

```
cspvpn:/opt/lissi/bin# ./make_pkcs10 -key <file> -out <file1>
-derout -subj <DN>
```

где

<code>-key &lt;file&gt;</code>	имя файла, в котором записана ключевая пара
<code>-out &lt;file1&gt;</code>	имя файла, в который будет записан запрос на локальный сертификат в формате PKCS#10
<code>-derout</code>	формирование запроса в формате DER (по умолчанию – PEM)
<code>-subj &lt;DN&gt;</code>	параметры для поля DistinguishedName (DN) сертификата.

Значение DN должно быть задано в виде `"/C=value0/L=value1/O=value2/..."`, Пробелам в `valueX` должна предшествовать обратная наклонная черта `'\'`.

Пример значения DN:

```
"/C=RU/L=Yubilejnyj/O=LISSI\ltd./OU=Info\System\Department/CN=Ivano
v"
```

Пример команды:

```
cspvpn:/opt/lissi/bin# ./make_pkcs10 -key pair_key -out
req_cer -subj "/C=RU/L=Yubilejnyj/O=LISSI\ltd."
```

**Шаг4** : проверьте созданный запрос, просмотрев файл при помощи утилиты `pkcs10_info`:

```
cspvpn:/opt/lissi/bin# ./pkcs10_info -in <file> -derin
```

где

<code>-in &lt;file&gt;</code>	файл с запросом
<code>-derin</code>	запрос представлен в формате DER (по умолчанию – PEM)

**Шаг5** : отправьте созданный запрос доступным вам способом на сервер доверенного Удостоверяющего Центра “ЛИССИ-УЦ/LISSICA”, где по данному запросу будет создан локальный сертификат.

**Шаг6** : получите из Удостоверяющего Центра СА и локальные сертификаты в виде файлов, а затем доставьте их любым способом на шлюз безопасности.

**Шаг7** : разместите в контейнере PKCS#12 доставленный локальный сертификат и файл с ключевой парой, используя утилиту `make_pkcs12`:

```
cspvpn:/opt/lissi/bin# ./make_pkcs12 -key <file> -cert <file>
-certinder -alias "<name>" -pass "<password>" -out <outfile>
```

где

<code>-key &lt;file&gt;</code>	входной файл с ключевой парой
<code>-cert &lt;file&gt;</code>	входной файл с сертификатом
<code>-certinder</code>	сертификат представлен в формате DER (по умолчанию - PEM)
<code>-alias "&lt;name&gt;"</code>	использовать 'name' в качестве имени ключевой пары (friendlyName)
<code>-pass "&lt;password&gt;"</code>	пароль для защиты контейнера
<code>-out &lt;outfile&gt;</code>	выходной файл контейнера PKCS#12

Пример команды:

```
cspvpn:/opt/lissi/bin# ./make_pkcs12 pair_key -cert  
loc_cer.crt -alias "gate1" -pass "12345"  
-out gate1_cont12.p12
```

**Шаг 8:** зарегистрируйте СА и локальный сертификаты в Продукте, используя утилиту `cert_mgr` из состава Продукта. Такая регистрация описана в документе [“Специализированные команды”](#).

## 13. Совместная работа на разных криптопровайдерах

Для совместной работы, например, трех 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 3.0", а на третьем - CSP VPN Gate со встроенной криптобиблиотекой "LirSSL", разработанной компанией "ЛИССИ", 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 Удостоверяющий Центр, имеющий CA сертификат, созданный с использованием криптопровайдера «КриптоПро CSP 3.0»
- для второго шлюза безопасности создать запрос с помощью криптопровайдера «КриптоПро CSP 3.0» и отослать его в УЦ с CA сертификатом этого же криптопровайдера
- на третьем шлюзе с помощью утилиты `make_key_pair` создать ключевую пару, выбрав для алгоритма подписи значение по умолчанию (ГОСТ Р 34.10-2001 в формате «Крипто-Про»), и создать запрос на сертификат с помощью утилиты `make_pkcs#10`
- созданный запрос отослать в Удостоверяющий Центр «ЛИССИ-УЦ/LISSICA».

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

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

## 14. 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>).