

# Отказоустойчивое решение с резервированием интерфейсов. Аутентификация на сертификатах, СКЗИ «КриптоПро CSP 2.0»

## Описание стенда

Сценарий иллюстрирует построение защищенного соединения между двумя подсетями, одна из которых - SN1-192.168.1.0/24 защищается шлюзом безопасности GW1, а вторая подсеть - SN2-192.168.2.0/24 защищается шлюзом безопасности GW2, имеющего две NIC в одном сетевом пространстве. Устройства IPHost1 и IPHost2 смогут общаться между собой по защищенному каналу (VPN). Все остальные соединения разрешены, но защищаться не будут. (Рисунок 1)

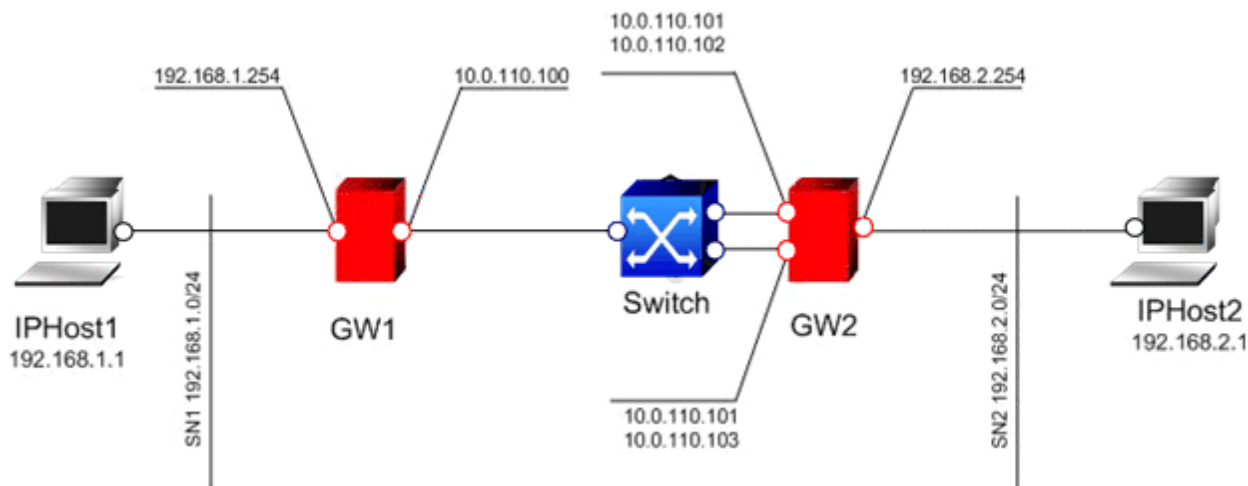


Рисунок 1

## Параметры CSP VPN Gate

Операционная система: SunOS 5.8

Версия и наименование продукта: CSP VPN GATE1000 build 2.2.7389

ПО криптопровайдера: «КриптоПро CSP» версии 2.0

## Параметры защищенного соединения

Параметры защищенного соединения между подсетями SN1 и SN2:

- Аутентификация на сертификатах
- IKE parameters:
  - Encryption algorithm – GOST
  - Hash algorithm – GOST

- . DH-group – group2 (1024)
- . IPsec parameters:
  - ESP encryption algorithm – GOST

## Логика работы отказоустойчивого решения

Основная идея состоит в использовании возможностей in.mpath-демона. Этот сервис позволяет иметь в системе несколько NIC в одном сегменте сети. Один интерфейс находится в рабочем состоянии, а второй – в «спящем». При пропадании линка, на активном интерфейсе автоматически активируется «спящий». При переключении, на запасной интерфейс переносятся настройки активного. После восстановления линка, происходит переключение на основной интерфейс. Система может пребывать в двух состояниях – normal и failover. В нашем примере интерфейс iprb0 резервируется интерфейсом iprb1. Весь трафик, в обоих состояниях идет через адрес 10.0.110.101.

## Настройка шлюза безопасности GW2

Сначала настроим устройство GW2. Все настройки будем производить удаленно, получив доступ к устройству по протоколу ssh с правами суперпользователя.

Для начала выполним несколько шагов по настройке in.mpath-демона.

1. Уменьшим значения таймаута для переключения. Для этого установим в `/etc/default/mpathd` значение `FAILURE_DETECTION_TIME` равное 2000.

```
#
#ident "@(#)mpathd.dfl 1.1 00/01/03 SMI"
#
# Time taken by mpathd to detect a NIC failure in ms. The minimum time
# that can be specified is 100 ms.
#
FAILURE_DETECTION_TIME=2000
#
# Failback is enabled by default. To disable failback turn off this option
#
FAILBACK=yes
#
# By default only interfaces configured as part of multipathing groups
# are tracked. Turn off this option to track all network interfaces
# on the system
#
TRACK_INTERFACES_ONLY_WITH_GROUPS=yes
```

2. Установим соответствие адресов псевдонимам интерфейсов. Исправления будем делать в `/etc/hosts`

```
#
# Internet host table
#
```

---

127.0.0.1	localhost	loghost
10.0.110.101	supp-gw-data	
10.0.110.102	supp-gw	
10.0.110.103	supp-gw-fail	
192.168.2.254	supp-gw-int	

### 3. Сделаем необходимые настройки в /etc/hostname.iprb0

```
gw2#vi /etc/hostname.iprb0
supp-gw group test -failover up \
addif supp-gw-data up
```

### 4. И в /etc/hostname.iprb1

```
gw2#vi /etc/hostname.iprb1
supp-gw-fail group test deprecated -failover standby up
```

## Регистрация CA сертификата

Для регистрации CA сертификата выполним следующие действия:

1. Доставим файл CA сертификата на шлюз безопасности. Для доставки можно воспользоваться утилитой `pscp.exe` из пакета Putty. Файл `ca.cer` доставим в предварительно созданный каталог `/certs`:

```
pscp ca.cer root@192.168.2.254:/certs
```



Предупреждение

**Среда передачи в этом случае должна быть доверенной**

2. С помощью утилиты `cert_mgr`, входящей в состав продукта, зарегистрируем сертификат в базе продукта:

```
gw2#/opt/VPNagent/bin/cert_mgr import -f /certs/ca.cer -t
```

## Регистрация локального сертификата

Для регистрации локального сертификата в базе продукта выполним следующие действия:

1. Доставим в файловую систему шлюза безопасности файл локального сертификата. Снова воспользуемся утилитой `pscp.exe`.

```
pscp local1.cer root@192.168.2.254:/certs
```

2. Доставим контейнеры с секретными ключами для локального сертификата на шлюз безопасности.

```
pscp -r local1.000 root@192.168.2.254: /var/CPROcsp/users
```

3. Получим имя контейнера с секретными ключами, для этого используем утилиту `csptest`, входящую в состав ПО криптопровайдера:

```
gw2# /opt/CPROcsp/src/csptest/csptest -keyset -machinekeyset -verifycontext
-enum_containers -unique
```

```
CryptAcquireContext succeeded.HCRYPTPROV: 1
```

```
HDIMAGE\\local1.000|
```

```
local1
```

```
OK.
```

4. Зарегистрируем локальный сертификат в базе продукта, используя утилиту `cert_mgr` из состава продукта. Импорт производим командой следующего вида:

```
gw2# /opt/VPNagent/bin/cert_mgr import -f /certs/local1.cer -kc
'HDIMAGE\\local1.000'
```

5. Убедимся, что сертификаты импортированы успешно:

```
gw2# /opt/VPNagent/bin/cert_mgr show
```

```
Found 2 certificates. No CRLs found.
```

```
1 Status: trusted 1.2.840.113549.1.9.1=support@s-
terra.com,C=RU,ST=Moscow,L=Zelenograd,O=S-Terra CSP,OU=Support
Group,CN=Support Group CA
```

```
2 Status: local 1.2.840.113549.1.9.1=support@s-
terra.com,C=RU,ST=Moscow,L=Zelenograd,O=S-Terra CSP,OU=Support
Group,CN=local1
```

## Создание политики безопасности

После регистрации сертификатов перейдем к созданию политики безопасности для GW2. Создавать политику будем в интерфейсе командной строки. Для входа в консоль перейдем в директорию `/opt/VPNagent/bin/` и запустим `cs_console`. Перейдем в режим настройки.

Зададим адрес шлюза по умолчанию:

```
gw2 (config)# ip route 0.0.0.0 0.0.0.0 10.0.110.100
```

Зададим параметры для IKE:

```
gw2 (config)#crypto isakmp policy 1
```

```
gw2 (config-isakmp)#hash md5
```

```
gw2 (config-isakmp)# encryption des
```

```
gw2 (config-isakmp)# authentication rsa-sig
```

```
gw2 (config-isakmp)# group 2
```

```
gw2 (config-isakmp) #exit
```

Создадим набор преобразований для IPsec:

```
gw2 (config) #crypto ipsec transform-set Gost esp-des
gw2 (cfg-crypto-trans) #mode tunnel
gw2 (cfg-crypto-trans) #exit
```

Опишем трафик, который планируется защищать. Для этого создадим расширенный список доступа.

```
gw2 (config) #ip access-list extended SN2toSN1
gw2 (config-ext-nacl) # permit ip 192.168.2.0 0.0.0.255 192.168.1.0 0.0.0.255
gw2 (config-ext-nacl) #exit
```

Создадим криптокарту:

```
gw2 (config) #crypto map CMAP 1 ipsec-isakmp
gw2 (config-crypto-map) # match address SN2toSN1
gw2 (config-crypto-map) # set transform-set Gost
gw2 (config-crypto-map) # set peer 10.0.110.100
gw2 (config-crypto-map) #exit
```

«Привяжем» криптокарту к интерфейсу, на котором будет терминироваться туннель:

```
gw2 (config) #interface FastEthernet0/0
gw2 (config-if) #crypto map CMAP
gw2 (config-if) #exit
```

Чтобы после переключения с основного интерфейса на резервный туннель перестроился, привяжем криптокарту к interface FastEthernet0/1:

```
gw2 (config) #interface FastEthernet0/1
gw2 (config-if) #crypto map CMAP
gw2 (config-if) #exit
```

Отключим обработку списка отозванных сертификатов (CRL):

```
gw2 (config) #crypto ca trustpoint s-terra_technological_trustpoint
gw2 (ca-trustpoint) # crl optional
gw2 (ca-trustpoint) #exit
```

Настройка устройства GW2 завершена. При выходе из конфигурационного режима происходит загрузка конфигурации.

В Приложении приведены [текст cisco-like конфигурации](#) , [текст LSP](#), [таблица маршрутов](#) и [вывод команды ifconfig -a](#)

## Настройка шлюза GW1

Регистрация CA и локального сертификатов производится так же, как и для шлюза безопасности GW2.

[Текст политики безопасности](#), [текст LSP](#), [таблица маршрутов](#) и [вывод команды ifconfig -a](#) для шлюза GW1 приведены в Приложении.

## Настройка устройства IPHost1

На устройстве IPHost1 в качестве шлюза по умолчанию нужно указать адрес внутреннего интерфейса шлюза безопасности GW1 - 192.168.1.254.

## Настройка устройства IPHost2

На устройстве IPHost2 в качестве шлюза по умолчанию нужно указать адрес внутреннего интерфейса устройства GW2 – 192.168.2.254.

## Проверка работоспособности стенда

После того, как настройка GW1и GW2 завершена, инициируем создание защищенного соединения.

1. На IPHost1 выполним команду:

```
ping 192.168.2.1
PING 192.168.2.1 (192.168.2.1): 56 data bytes
64 bytes from 192.168.2.1: icmp_seq=0 ttl=61 time=5.8 ms
64 bytes from 192.168.2.1: icmp_seq=1 ttl=61 time=1.6 ms
64 bytes from 192.168.2.1: icmp_seq=2 ttl=61 time=1.5 ms
64 bytes from 192.168.2.1: icmp_seq=3 ttl=61 time=1.5 ms
64 bytes from 192.168.2.1: icmp_seq=4 ttl=61 time=1.6 ms
64 bytes from 192.168.2.1: icmp_seq=5 ttl=61 time=1.6 ms
64 bytes from 192.168.2.1: icmp_seq=6 ttl=61 time=2.7 ms
64 bytes from 192.168.2.1: icmp_seq=7 ttl=61 time=1.6 m
```

В результате выполнения этой команды между устройствами GW1 и GW2 будет установлен VPN туннель. Убедиться в этом можно при помощи утилиты `sa_show`.

2. На устройстве GW2 выполним команду:

```
gw2#/opt/VPNagent/bin/sa_show
```

```
IPSec SA Num (Remote Addr,Port)-(Local Addr,Port) Protocol Action Type Sent
Rec
IPSec SA 1 8 (192.168.1.0-192.168.1.255,*)-(192.168.2.0-192.168.2.255,*) *
ESP tunn 8951 11904
```

Теперь, не останавливая посылку ICMP-пакетов с IPHost1, отключим основной интерфейс на GW2. Дождемся установления SA через резервный iprb1 интерфейс и возобновления связи:

```
64 bytes from 192.168.2.1: icmp_seq=365 ttl=61 time=1.5 ms
64 bytes from 192.168.2.1: icmp_seq=366 ttl=61 time=1.5 ms
64 bytes from 192.168.2.1: icmp_seq=367 ttl=61 time=1.5 ms
64 bytes from 192.168.2.1: icmp_seq=368 ttl=61 time=1.6 ms

64 bytes from 192.168.2.1: icmp_seq=372 ttl=61 time=3446.6 ms
64 bytes from 192.168.2.1: icmp_seq=373 ttl=61 time=2447.0 ms
64 bytes from 192.168.2.1: icmp_seq=375 ttl=61 time=447.7 ms
64 bytes from 192.168.2.1: icmp_seq=376 ttl=61 time=2.8 ms
64 bytes from 192.168.2.1: icmp_seq=377 ttl=61 time=2.7 ms
```

Убедиться в том, что туннель перестроился, можно при помощи утилиты `sa_show`.

### 3. На устройстве GW2 выполним команду:

```
gw2#/opt/VPNagent/bin/sa_show
```

```
IPSec SA Num (Remote Addr,Port)-(Local Addr,Port) Protocol Action Type Sent
Rec
IPSec SA 1 6 (192.168.1.0-192.168.1.255,*)-(192.168.2.0-192.168.2.255,*) *
ESP tunn 10238 15896
```

После этого вернем стэнд в исходное состояние и наблюдаем переключение с резервного на основной интерфейс:

```
64 bytes from 192.168.2.1: icmp_seq=424 ttl=61 time=1.5 ms
64 bytes from 192.168.2.1: icmp_seq=425 ttl=61 time=2.7 ms
64 bytes from 192.168.2.1: icmp_seq=426 ttl=61 time=1.5 ms
64 bytes from 192.168.2.1: icmp_seq=427 ttl=61 time=1.5 ms

64 bytes from 192.168.2.1: icmp_seq=8 ttl=252 time=4228.4 ms
64 bytes from 192.168.2.1: icmp_seq=9 ttl=252 time=3229.0 ms
64 bytes from 192.168.2.1: icmp_seq=10 ttl=252 time=2229.4 ms
64 bytes from 192.168.2.1: icmp_seq=13 ttl=252 time=20.1 ms
64 bytes from 192.168.2.1: icmp_seq=14 ttl=252 time=2.0 ms
```

Можно убедиться, что SA опять был перестроен. Для этого на GW2 выполним команду:

```
gw2#/opt/VPNagent/bin/sa_show

IPSec SA Num (Remote Addr,Port)-(Local Addr,Port) Protocol Action Type Sent
Rec
IPSec SA 1 13 (192.168.103.0-192.168.103.255,*)-(10.0.110.0-10.0.110.255,*)
* ESP tunn 13685 18547
```

Мы только что создали конфигурацию, обеспечивающую взаимодействие между устройствами из подсетей SN1и SN2 по защищенному каналу, с использованием резервирования интерфейсов на шлюзе CSP VPN Gate (GW2).

## Приложение

### Текст cisco-like конфигурации для GW1

```
crypto ipsec df-bit copy
crypto isakmp identity dn
crypto isakmp keepalive 10 5
username cscons password csp
hostname GW1
enable password csp
!
logging trap debugging
!
crypto isakmp policy 1
  hash md5
  encryption des
  authentication rsa-sig
  group 2
!
crypto ipsec transform-set Gost esp-des
  mode tunnel
!
ip access-list extended SN1toSN2
  permit ip 192.168.1.0 0.0.0.255 192.168.2.0 0.0.0.255
!
crypto map CMAP 1 ipsec-isakmp
  match address SN1toSN2
  set transform-set Gost
  set pfs group2
  set peer 10.0.110.101
!
interface FastEthernet0/0
```

```
ip address 10.0.110.100 255.0.0.0
crypto map CMAP
interface FastEthernet0/1
ip address 192.168.1.254 255.255.255.0
!
crypto ca trustpoint s-terra_technological_trustpoint
crl optional
crypto CA certificate chain s-terra_technological_trustpoint
certificate 551C69033C5A62864FCD2D961858B88B
3082036730820314A003====skip====9E0C4C06767E9824AAC65993BB3F9F918325EA384EB
E
quit

ip route 0.0.0.0 0.0.0.0 10.0.110.101 1
end
```

## Текст LSP для устройства GW1

```
# This is automatically generated LSP
#
# Conversion Date/Time: Fri Feb 15 11:45:33 2008

GlobalParameters(
  Title = "This LSP was automatically generated by
CSP Converter at Fri Feb 15 11:45:33 2008"
  Version = "2.1"
  CRLHandlingMode = OPTIONAL
  LDAPLogMessageLevel = DEBUG
  SystemLogMessageLevel = DEBUG
  PolicyLogMessageLevel = DEBUG
  CertificatesLogMessageLevel = DEBUG
)

SyslogSettings(
  Server = 127.0.0.1
  Facility = LOG_LOCAL7
)

RoutingTable(
  Routes *=
    Route(
      Destination = 0.0.0.0/0
      Gateway = 10.0.110.101
      Metric = 1
    )
  )
)
```

```
IKETransform IKETransform_1
(
    CipherAlg      *= "G2814789CPR01-K256-CBC-65534"
    HashAlg        *= "GR341194CPR01-65534"
    GroupID        *= MODP_1024
    LifetimeSeconds = 86400
)

ESPProposal ESP_Gost
(
    Transform* = ESPTransform
    (
        CipherAlg*      = "G2814789CPR01-K256-CBC-254"
        LifetimeSeconds = 3600
        LifetimeKilobytes = 4608000
    )
)

CertDescription ca
(
    Issuer          *= "1.2.840.113549.1.9.1=support@s-
terra.com,C=RU,ST=Moscow,L=Zelenograd,O=S-Terra CSP,OU=Support
Group,CN=Support Group CA"
    SerialNumber    = "551c69033c5a62864fcd2d961858b88b"
    Subject          *= "1.2.840.113549.1.9.1=support@s-
terra.com,C=RU,ST=Moscow,L=Zelenograd,O=S-Terra CSP,OU=Support
Group,CN=Support Group CA"
)

AuthMethodGOSTSign auth_ca
(
    LocalID          = IdentityEntry( DistinguishedName* = USER_SPECIFIC_DATA
)
    DoNotMapRemoteIDToCert = TRUE
    AcceptCredentialFrom    *= ca
    SendRequestMode         = ALWAYS
    SendCertMode            = ALWAYS
)

IKERule IKE_CMAP_1
(
    Transform* = IKETransform_1
    AggrModeAuthMethod *= auth_ca
    MainModeAuthMethod *= auth_ca
    DoAutopass         = TRUE
)
```

```

        DPDIdleDuration      = 10
        DPDResponseDuration = 5
        DPDRetries           = 5
    )

IPsecAction CMAP_1
(
    TunnelingParameters *= TunnelEntry(
        PeerIPAddress = 10.0.110.101

        DFHandling=COPY
    )
    ContainedProposals *= ( ESP_Gost )
    GroupID *= MODP_1024
    IKERule = IKE_CMAP_1
)

FilteringRule Filter_nil_acl_CMAP_1
(
    LocalIPFilter *= FilterEntry( IPAddress *= 192.168.1.0/24 )
    PeerIPFilter  *= FilterEntry( IPAddress *= 192.168.2.0/24 )
    NetworkInterfaces *= "rtls0"
    Action *= ( CMAP_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 *= "rtls0"
    Action *= ( PASS )
)

FilteringRule Filter_nil_acl_1
(
    LocalIPFilter *= FilterEntry( IPAddress *= 0.0.0.0..255.255.255.255 )
    PeerIPFilter  *= FilterEntry( IPAddress *= 0.0.0.0..255.255.255.255 )
    NetworkInterfaces *= "rtls1"
    Action *= ( PASS )
)

```

## Текст cisco-like конфигурации для GW2

```

crypto ipsec df-bit copy
crypto isakmp identity dn

```

```
username cscons password csp
hostname suppgw2
enable password csp
!
logging trap debugging
!
crypto isakmp policy 1
  hash md5
  encryption des
  authentication rsa-sig
  group 2
!
crypto ipsec transform-set Gost esp-des
  mode tunnel
!
ip access-list extended SN2toSN1
  permit ip 192.168.2.0 0.0.0.255 192.168.1.0 0.0.0.255
!
crypto map CMAP 1 ipsec-isakmp
  match address SN2toSN1
  set transform-set Gost
  set pfs group2
  set peer 10.0.110.100
!
interface FastEthernet0/0
  ip address 10.0.110.102 255.255.255.0
  ip address 10.0.110.101 255.255.255.0 secondary
  crypto map CMAP
interface FastEthernet0/1
  ip address 10.0.110.103 255.255.255.0
  crypto map CMAP
interface FastEthernet0/2
  ip address 192.168.2.254 255.255.255.0
!
crypto ca trustpoint s-terra_technological_trustpoint
  crl optional
crypto CA certificate chain s-terra_technological_trustpoint
certificate 551C69033C5A62864FCD2D961858B88B
3082036730820314A00====skip====0C4C06767E9824AACC65993BB3F9F918325EA384EBE
quit
!
ip route 0.0.0.0 0.0.0.0 10.0.110.100 1
end
```

## Текст LSP для устройства GW2

```
# This is automatically generated LSP
#
# Conversion Date/Time: Thu Feb 14 11:48:45 2008

GlobalParameters(
    Title = "This LSP was automatically generated by
CSP Converter at Thu Feb 14 11:48:45 2008"
    Version = "2.1"
    CRLHandlingMode = OPTIONAL
    LDAPLogMessageLevel = DEBUG
    SystemLogMessageLevel = DEBUG
    PolicyLogMessageLevel = DEBUG
    CertificatesLogMessageLevel = DEBUG
)

SyslogSettings(
    Server = 127.0.0.1
    Facility = LOG_LOCAL7
)

RoutingTable(
    Routes *=
        Route(
            Destination = 0.0.0.0/0
            Gateway = 10.0.110.100
            Metric = 1
        )
)

IKETransform IKETransform_1
(
    CipherAlg *= "G2814789CPR01-K256-CBC-65534"
    HashAlg *= "GR341194CPR01-65534"
    GroupID *= MODP_1024
    LifetimeSeconds = 86400
)

ESPProposal ESP_Gost
(
    Transform* = ESPTransform
    (
        CipherAlg* = "G2814789CPR01-K256-CBC-254"
        LifetimeSeconds = 3600
    )
)
```

```
        LifetimeKilobytes    = 4608000
    )
)

CertDescription ca
(
    Issuer                  *= "1.2.840.113549.1.9.1=support@s-
terra.com,C=RU,ST=Moscow,L=Zelenograd,O=S-Terra CSP,OU=Support
Group,CN=Support Group CA"
    SerialNumber           = "551c69033c5a62864fcd2d961858b88b"
    Subject                 *= "1.2.840.113549.1.9.1=support@s-
terra.com,C=RU,ST=Moscow,L=Zelenograd,O=S-Terra CSP,OU=Support
Group,CN=Support Group CA"
)

AuthMethodGOSTSign auth_ca
(
    LocalID                = IdentityEntry( DistinguishedName* = USER_SPECIFIC_DATA
)

    DoNotMapRemoteIDToCert = TRUE
    AcceptCredentialFrom    *= ca
    SendRequestMode         = ALWAYS
    SendCertMode            = ALWAYS
)

IKERule IKE_CMAP_1
(
    Transform* = IKETransform_1
    AggrModeAuthMethod *= auth_ca
    MainModeAuthMethod *= auth_ca
    DoAutopass        = TRUE
    DoNotUseDPD       = TRUE
)

IPsecAction CMAP_1
(
    TunnelingParameters *= TunnelEntry(
        PeerIPAddress = 10.0.110.100

        DFHandling=COPY
    )
    ContainedProposals *= ( ESP_Gost )
    GroupID *= MODP_1024
    IKERule = IKE_CMAP_1
)

FilteringRule Filter_nil_acl_CMAP_1
```

```
(
    LocalIPFilter *= FilterEntry( IPAddress *= 192.168.2.0/24 )
    PeerIPFilter  *= FilterEntry( IPAddress *= 192.168.1.0/24 )
    NetworkInterfaces *= "iprb0"
    Action *= ( CMAP_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 *= "iprb0"
    Action *= ( PASS )
)

AuthMethodGOSTSign auth_ca_1
(
    LocalID          = IdentityEntry( DistinguishedName* = USER_SPECIFIC_DATA
)
    DoNotMapRemoteIDToCert = TRUE
    AcceptCredentialFrom   *= ca
    SendRequestMode        = ALWAYS
    SendCertMode           = ALWAYS
)

IKERule IKE_CMAP_1_1
(
    Transform* = IKETransform_1
    AggrModeAuthMethod *= auth_ca_1
    MainModeAuthMethod *= auth_ca_1
    DoAutopass          = TRUE
    DoNotUseDPD         = TRUE
)

IPsecAction CMAP_1_1
(
    TunnelingParameters *= TunnelEntry(
        PeerIPAddress = 10.0.110.100

        DFHandling=COPY
    )
    ContainedProposals *= ( ESP_Gost )
    GroupID *= MODP_1024
    IKERule = IKE_CMAP_1_1
)

```

```

FilteringRule Filter_nil_acl_CMAP_1_1
(
    LocalIPFilter *= FilterEntry( IPAddress *= 192.168.2.0/24 )
    PeerIPFilter  *= FilterEntry( IPAddress *= 192.168.1.0/24 )
    NetworkInterfaces *= "iprb1"
    Action *= ( CMAP_1_1 )
)

FilteringRule Filter_nil_acl_1
(
    LocalIPFilter *= FilterEntry( IPAddress *= 0.0.0.0..255.255.255.255 )
    PeerIPFilter  *= FilterEntry( IPAddress *= 0.0.0.0..255.255.255.255 )
    NetworkInterfaces *= "iprb1"
    Action *= ( PASS )
)

FilteringRule Filter_nil_acl_2
(
    LocalIPFilter *= FilterEntry( IPAddress *= 0.0.0.0..255.255.255.255 )
    PeerIPFilter  *= FilterEntry( IPAddress *= 0.0.0.0..255.255.255.255 )
    NetworkInterfaces *= "iprb2"
    Action *= ( PASS )
)

```

## Таблица маршрутизации GW1

```
gw1#netstat -rn
```

```
Routing Table: IPv4
```

Destination	Gateway	Flags	Ref	Use	Interface
192.168.1.0	192.168.1.254	U	1	20	rtls1
10.0.0.0	10.0.110.100	U	1	98	rtls0
default	10.0.110.101	UG	1	18	
127.0.0.1	127.0.0.1	UH	2	2016	lo0

## Таблица маршрутизации GW2

```
gw2#netstat -rn
```

```
Routing Table: IPv4
```

Destination	Gateway	Flags	Ref	Use	Interface
10.0.110.0	10.0.110.102	U	1	40	iprb0

---

10.0.110.0	10.0.110.101	U	1	0	iprb0:1
10.0.110.0	10.0.110.102	U	1	1	iprb1
192.168.2.0	192.168.2.254	U	1	8	iprb2
default	10.0.110.100	UG	1	0	
127.0.0.1	127.0.0.1	UH	2	242	lo0

## Вывод команды `ifconfig -a` устройства GW1

```
gw1#ifconfig -a
lo0: flags=1000849<UP,LOOPBACK,RUNNING,MULTICAST,IPv4> mtu 8232 index 1
    inet 127.0.0.1 netmask ff000000
rtls0: flags=1000843<UP,BROADCAST,RUNNING,MULTICAST,IPv4> mtu 1500 index 8
    inet 10.0.110.100 netmask ff000000 broadcast 10.255.255.255
    ether 0:50:fc:90:7f:1c
rtls1: flags=1000843<UP,BROADCAST,RUNNING,MULTICAST,IPv4> mtu 1500 index 9
    inet 192.168.1.254 netmask ffffffff broadcast 192.168.1.255
    ether 4c:0:10:71:5c:b6
```

## Вывод команды `ifconfig -a` устройства GW2

Стартовое состояние:

```
gw2#ifconfig -a
lo0: flags=1000849<UP,LOOPBACK,RUNNING,MULTICAST,IPv4> mtu 8232 index 1
    inet 127.0.0.1 netmask ff000000
iprb0: flags=9000843<UP,BROADCAST,RUNNING,MULTICAST,IPv4,NOFAILOVER> mtu
1500 index 2
    inet 10.0.110.102 netmask ffffffff broadcast 10.0.110.255
    groupname test
    ether 0:a0:c9:2b:7d:f2
iprb0:1: flags=1000843<UP,BROADCAST,RUNNING,MULTICAST,IPv4> mtu 1500 index 2
    inet 10.0.110.101 netmask ffffffff broadcast 10.0.110.255
iprb1:
flags=69040843<UP,BROADCAST,RUNNING,MULTICAST,DEPRECATED,IPv4,NOFAILOVER,STA
NDBY,INACTIVE> mtu 1500 index 3
    inet 10.0.110.103 netmask ffffffff broadcast 10.0.110.255
    groupname test
    ether 0:90:27:9b:b0:cf
iprb2: flags=1000843<UP,BROADCAST,RUNNING,MULTICAST,IPv4> mtu 1500 index 4
    inet 192.168.2.254 netmask ffffffff broadcast 192.168.2.255
    ether 0:a0:c9:85:6f:17
```

После отказа основного линка на `iprb0`, будет «поднят» виртуальный интерфейс с адресом 10.0.110.101 на `iprb1`.

```
gw2#ifconfig -a
lo0: flags=1000849<UP,LOOPBACK,RUNNING,MULTICAST,IPv4> mtu 8232 index 1
    inet 127.0.0.1 netmask ff000000
```

---

```
iprb0: flags=19000843<UP,BROADCAST,RUNNING,MULTICAST,IPv4,NOFAILOVER,FAILED>
mtu 1500 index 2
    inet 10.0.110.102 netmask ffffffff broadcast 10.0.110.255
    groupname test
    ether 0:a0:c9:2b:7d:f2

iprb1:
flags=29040843<UP,BROADCAST,RUNNING,MULTICAST,DEPRECATED,IPv4,NOFAILOVER,STA
NDBY> mtu 1500 index 3
    inet 10.0.110.103 netmask ffffffff broadcast 10.0.110.255
    groupname test
    ether 0:90:27:9b:b0:cf

iprb1:1: flags=21000843<UP,BROADCAST,RUNNING,MULTICAST,IPv4,STANDBY> mtu
1500 index 3
    inet 10.0.110.101 netmask ffffffff broadcast 10.0.110.255

iprb2: flags=1000843<UP,BROADCAST,RUNNING,MULTICAST,IPv4> mtu 1500 index 4
    inet 192.168.2.254 netmask ffffffff broadcast 192.168.2.255
    ether 0:a0:c9:85:6f:17
```