Quick Note 62

IKEv2 IPsec VPN from TransPort WR to StrongSwan using Certificates

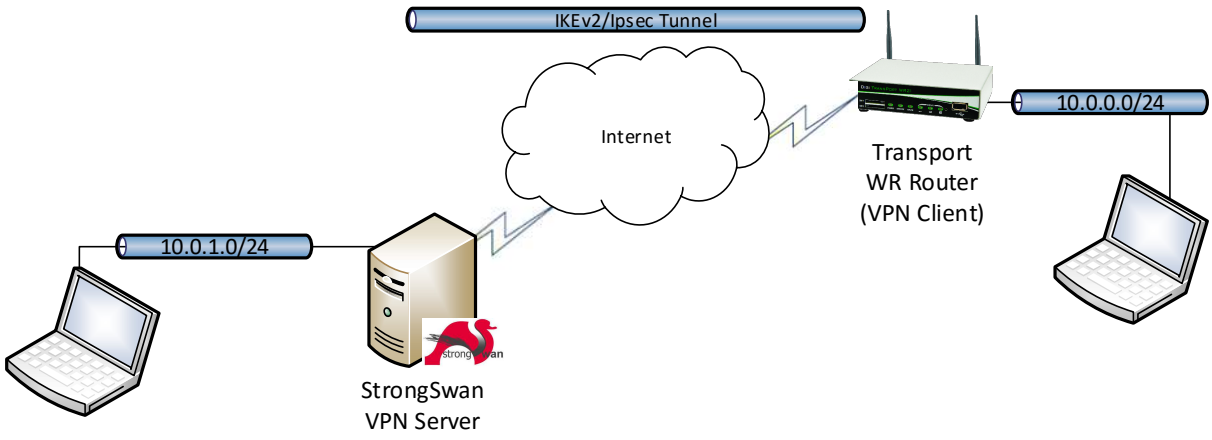
Contents

1 INTRODUCTION

1.1 Outline

This document describes how to configure a VPN IPsec tunnel between a Digi TransPort WR to a StrongSwan server IKEv2 using Certificates authentication.

The network diagram considered in this example is the following:



This guide details the steps involved in configuring a Digi TransPort router to act as an IPsec VPN client to a StrongSwan appliance configured as an IPsec VPN server using IKEv2 and Certificates authentication.

1.2 Assumptions

This guide has been written for technically competent personnel who are able to configure a standard IPSec tunnel between 2 TransPort WR routers and are familiar with the use of routing protocols.

This application note applies to:

Model: Digi TransPort WR11/21/31/41/44

Firmware versions:

WR21: 5.2.17.10 and later

Configuration: This document assumes that the devices are set to their factory default configurations. Most configuration commands are shown only if they differ from the factory default.

1.3 Corrections

Requests for corrections or amendments to this application note are welcome and should be addressed to: tech.support@digi.com.

Requests for new application notes can be sent to the same address.

1.4 Version

Version Number	Status
1.0	Published

2 STRONGSWAN INSTALLATION AND CERTIFICATES CREATION

2.1 Installing StrongSwan

The following instructions refer to an Ubuntu sysetm for the server side, please refer to http://www.strongswan.org for installation on other Linux distribution.

From the terminal, issue the command “sudo apt-get install strongswan”:

```
digi@Digi:~$ sudo apt-get install strongswan
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following packages were automatically installed and are no longer required:
  accountsservice-ubuntu-schemas accountsservice-ubuntu-touch-schemas address-book-service
biometryd-bin evolution-data-server-utouch folks-common history-service indicator-transfer
  indicator-transfer-download-manager libbiometryd1 libboost-system1.61.0 libcgmanager0 libclick-
0.4-0 libconnectivity-qt1 libdbus-cpp5 libfolks-eds25 libfolks25 libhistoryservice0
  libindicator-transfer0 libleveldb1v5 liblightdm-qt5-3-0 libmediascanner-2.0-4 libmiral2
libmission-control-plugins0 libonline-accounts-daemon1 libonline-accounts-qt1 libpoppler-qt5-1
libqdjango-db0
  libqgsttools-p1 libqmenumodel0 libqofono-qt5-0 libqt5contacts5 libqt5multimedia5-plugins
libqt5multimediaquick-p5 libqt5multimediawidgets5 libqt5sensors5 libqt5versit5 libqt5xmlpatterns5
libsnappy1v5
  libsystemsettings1 libtelepathy-qt4-2 libtelepathy-qt5-0 libthumbnailer-qt1.0 libtrust-store2
libubuntu-location-service3 libusermetricsinput1 libusermetricsoutput1 linux-headers-4.10.0-19
  linux-headers-4.10.0-19-generic linux-headers-4.10.0-32 linux-headers-4.10.0-32-generic linux-
image-4.10.0-19-generic linux-image-4.10.0-32-generic linux-image-extra-4.10.0-19-generic
  linux-image-extra-4.10.0-32-generic mediascanner2.0 mir-client-platform-mesa5 mir-graphics-
drivers-desktop mir-platform-graphics-mesa-kms12 mir-platform-graphics-mesa-x12 mir-platform-
input-evdev6
  policykit-unity8 qmenumodel-qml qml-module-biometryd qml-module-ofono qml-module-
pamauthentication0.1 qml-module-qmltermwidget1.0 qml-module-qtmultimedia qml-module-qtqml-
statemachine
  qml-module-qtquick-xmllistmodel qml-module-ubuntu-connectivity qml-module-ubuntu-
onlineaccounts2 qml-module-ubuntu-settings-components qml-module-ubuntu-thumbnailer0.1
qtcontact5-galera
  qtdeclarative5-gsettings1.0 qtdeclarative5-qtmir-plugin qtdeclarative5-ubuntu-settings-
components qtdeclarative5-ubuntu-telephony0.1 qtdeclarative5-unity-notifications-plugin qtmir-
desktop
  qtubuntu-appmenutheme qtubuntu-desktop qtubuntu-print sqlite3 telepathy-mission-control-5
telephony-service thumbnailer-service tone-generator ubuntu-application-api3-desktop ubuntu-
keyboard-data
  ubuntu-printing-app ubuntu-terminal-app unity-plugin-scopes unity-scope-mediascanner2 unity8-
common unity8-private upstart usermetricsservice
Use 'sudo apt autoremove' to remove them.
The following NEW packages will be installed:
  strongswan
0 upgraded, 1 newly installed, 0 to remove and 115 not upgraded.
Need to get 26,8 kB of archives.
After this operation, 175 kB of additional disk space will be used.
Get:1 http://de.archive.ubuntu.com/ubuntu zesty-updates/main amd64 strongswan all 5.5.1-
1ubuntu3.2 [26,8 kB]
Fetched 26,8 kB in 0s (80,2 kB/s)
Selecting previously unselected package strongswan.
(Reading database ... 273320 files and directories currently installed.)
Preparing to unpack .../strongswan_5.5.1-1ubuntu3.2_all.deb ...
Unpacking strongswan (5.5.1-1ubuntu3.2) ...
Setting up strongswan (5.5.1-1ubuntu3.2) ...
```

NOTE: The StrongSwan software is now installed and ready to use. In this document, the IPsec VPN will use Certificates authentication. If the set of certificates/keys is already available, you can skip until section 3.

2.2 Installing the PKI tool

In this example the StrongSwan PKI tool will be used to generate the Certificates and Keys needed for the authentication of the tunnel. This tool can be installed using the command “sudo apt install strongswan-pki”:

```
digi@Digi:~$ sudo apt install strongswan-pki
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following packages were automatically installed and are no longer required:
  accountsservice-ubuntu-schemas accountsservice-ubuntu-touch-schemas address-book-service
biometryd-bin evolution-data-server-utouch folks-common history-service indicator-transfer
  indicator-transfer-download-manager libbiometryd1 libboost-system1.61.0 libcgmanager0 libclick-
0.4-0 libconnectivity-qt1 libdbus-cpp5 libfolks-eds25 libfolks25 libhistoryservice0
  libindicator-transfer0 libleveldb1v5 liblightdm-qt5-3-0 libmediascanner-2.0-4 libmiral2
libmission-control-plugins0 libonline-accounts-daemon1 libonline-accounts-qt1 libpoppler-qt5-1
libqdjango-db0
  libqgsttools-p1 libqmenumodel0 libqofono-qt5-0 libqt5contacts5 libqt5multimedia5-plugins
libqt5multimediaquick-p5 libqt5multimediawidgets5 libqt5sensors5 libqt5versit5 libqt5xmlpatterns5
libsnappy1v5
  libsystemsettings1 libtelepathy-qt4-2 libtelepathy-qt5-0 libthumbnailer-qt1.0 libtrust-store2
libubuntu-location-service3 libusermetricsinput1 libusermetricsoutput1 linux-headers-4.10.0-19
  linux-headers-4.10.0-19-generic linux-headers-4.10.0-32 linux-headers-4.10.0-32-generic linux-
image-4.10.0-19-generic linux-image-4.10.0-32-generic linux-image-extra-4.10.0-19-generic
  linux-image-extra-4.10.0-32-generic mediascanner2.0 mir-client-platform-mesa5 mir-graphics-
drivers-desktop mir-platform-graphics-mesa-kms12 mir-platform-graphics-mesa-x12 mir-platform-
input-evdev6
  policykit-unity8 qmenumodel-qml qml-module-biometryd qml-module-ofono qml-module-
pamauthentication0.1 qml-module-qmltermwidget1.0 qml-module-qtmultimedia qml-module-qtqml-
statemachine
  qml-module-qtquick-xmllistmodel qml-module-ubuntu-connectivity qml-module-ubuntu-
onlineaccounts2 qml-module-ubuntu-settings-components qml-module-ubuntu-thumbnailer0.1
qtcontact5-galera
  qtdeclarative5-gsettings1.0 qtdeclarative5-qtmir-plugin qtdeclarative5-ubuntu-settings-
components qtdeclarative5-ubuntu-telephony0.1 qtdeclarative5-unity-notifications-plugin qtmir-
desktop
  qtubuntu-appmenutheme qtubuntu-desktop qtubuntu-print sqlite3 telepathy-mission-control-5
telephony-service thumbnailer-service tone-generator ubuntu-application-api3-desktop ubuntu-
keyboard-data
  ubuntu-printing-app ubuntu-terminal-app unity-plugin-scopes unity-scope-mediascanner2 unity8-
common unity8-private upstart usermetricsservice
Use 'sudo apt autoremove' to remove them.
The following NEW packages will be installed:
  strongswan-pki
0 upgraded, 1 newly installed, 0 to remove and 115 not upgraded.
Need to get 139 kB of archives.
After this operation, 776 kB of additional disk space will be used.
Get:1 http://de.archive.ubuntu.com/ubuntu zesty-updates/universe amd64 strongswan-pki amd64
5.5.1-1ubuntu3.2 [139 kB]
Fetched 139 kB in 0s (409 kB/s)
Selecting previously unselected package strongswan-pki.
(Reading database ... 273305 files and directories currently installed.)
Preparing to unpack .../strongswan-pki_5.5.1-1ubuntu3.2_amd64.deb ...
Unpacking strongswan-pki (5.5.1-1ubuntu3.2) ...
Setting up strongswan-pki (5.5.1-1ubuntu3.2) ...
Processing triggers for man-db (2.7.6.1-2) ...
```

2.3 Create a Root CA Certificate

First the CA key is created:

```
digi@Digi:~$  ipsec pki --gen --outform pem > caKey.pem
```

Where “cakey.pem” will be the name of the key that will be created.

With that, the CA certificates can be created:

```
digi@Digi:~$ ipsec pki --self --in caKey.pem --dn "C=DE, O=Digi, CN=digiCA" --ca --
outform pem > caCert.pem
```

Where:

Parameter	Description
caKey.pem	CA Key just created
C=DE	The two-letter ISO 3166 abbreviation for your country.
O=Digi	The exact legal name of your organization. Do not abbreviate your organization name.
CN=digiCA	Common Name for the CA
caCert.pem	Name of the CA certificate that will be created

The CA certificate just created, has to be moved to the **/etc/ipsec.d/cacerts** folder, in order to be cprrectlòy used by StrongSwan:

```
digi@Digi:~$ sudo cp caCert.pem /etc/ipsec.d/cacerts
```

2.4 Create a CA-Signed Server Certificate

The private Key for the Server is created with the following command:

```
digi@Digi:~$ ipsec pki --gen --outform pem > SrvKey.pem
```

The Server key just created, has to be moved to the **/etc/ipsec.d/private** folder, in order to be cprrectly used by StrongSwan:

```
digi@Digi:~$ sudo cp SrvKey.pem /etc/ipsec.d/private
```

Then, the Server Certificate can be created:

```
digi@Digi:~$ ipsec pki --pub --in SrvKey.pem | ipsec pki --issue --cacert caCert.pem --
cakey caKey.pem --dn "C=DE,O=Digi, CN=server" --flag serverAuth --flag ikeIntermediate
--san server --outform pem >  SrvCert.pem
```

Where:

Parameter	Description
SrvKey.pem	Server Key just created
caCert.pem	CA certificate created in previous step
caKey.pem	CA key created in previous step
C=DE	The two-letter ISO 3166 abbreviation for your country.
O=Digi	The exact legal name of your organization. Do not abbreviate your organization name.
CN=server	Common Name for the Server
SrvCert.pem	Name of the Server certificate that will be created

The Server Certificate just created, has to be moved to the **/etc/ipsec.d/certs** folder, in order to be correctly used by StrongSwan:

```
digi@Digi:~$ sudo cp SrvCert.pem /etc/ipsec.d/certs
```

2.5 Create a CA-Signed Client Certificate

The private Key for the Client is created with the following command:

```
digi@Digi:~$ ipsec pki --gen --outform pem > CliKey.pem
```

Then, the Client Certificate can be created:

```
digi@Digi:~$ ipsec pki --pub --in CliKey.pem | ipsec pki --issue --cacert caCert.pem --
cakey caKey.pem --dn "C=DE,O=Digi, CN=client" --san client --outform pem >
ClientCert.pem
```

Where:

Parameter	Description
CliKey.pem	Server Key just created
caCert.pem	CA certificate created in previous step
caKey.pem	CA key created in previous step
C=DE	The two-letter ISO 3166 abbreviation for your country.
O=Digi	The exact legal name of your organization. Do not abbreviate your organization name.
CN=client	Common Name for the Server
ClientCert.pem	Name of the Server certificate that will be created

3 STRONGSWAN CONFIGURATION

3.1 LAN Interface

Configure the Local interface for the StrongSwan Server. In this example, the Ethernet interface used for LAN is called enp0s8:

```
digi@Digi:~$ sudo ifconfig enp0s8 10.0.1.100
digi@Digi:~$ sudo ifconfig enp0s8 netmask 255.255.255.0
```

3.2 WAN Interface

The ETH Interface used for WAN is named enp0s3 and is configured as a DHCP Client.

In order to check which IP is assigned the command “ifconfig” can be used:

```
digi@Digi:~$ ifconfig enp0s3
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.104.1.125  netmask 255.255.255.0  broadcast 10.104.1.255
        inet6 fe80::588c:91d1:d0f7:a888  prefixlen 64  scopeid 0x20<link>
        ether 08:00:27:e7:99:d4  txqueuelen 1000  (Ethernet)
        RX packets 287564  bytes 28927717 (28.9 MB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 10841  bytes 1086181 (1.0 MB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

3.3 Uploading Certificates and Keys

Once you have the set of Certificates and Keys, that must be transferred to the system where the StronSwan VPN server runs.

If the certificates have been created following section 2, all the files are already in the correct folders.

If not, be sure that the files are uploaded on the server as following:

File	Description	Folder
caCert.pem	CA certificate	/etc/ipsec.d/cacerts
SrvKey.pem	Server Key	/etc/ipsec.d/private
SrvCert.pem	Server Certificate	/etc/ipsec.d/certs

3.4 IKEv2/IPsec Tunnel Configuration

The IPsec configuration used by StrongSwan, is contained in two main configuration files:

- ipsec.conf : Used for Phase 1 (IKE) and Phase 2 IPsec configuration

- ipsec.secrets : Used for many types of secretes. In our example, it defines the Server Private Key to use.

3.4.1 ipsec.conf

Edit the ipsec.conf file using a text editor (nano is used in this example):

```
# ipsec.conf - strongSwan IPsec configuration file
# basic configuration

config setup

# Add connections here.


conn %default
    keyexchange=ikev2
    ike=aes256-sha1-modp1024
    esp=aes256-sha1
    dpdaction=clear
    dpddelay=300s

conn DigiWR
    left=%any
    leftcert=SrvCert.pem
    leftsubnet=10.0.1.0/24
    right=%any
    rightsubnet=10.0.0.0/24
    auto=add
```

Where:

Parameter	Value	Description
	conn %default	
keyexchange	Ikev2	This tells Strongswan to use Ikev2. This parameter is used by default in strongswan 5.x
Ike	aes256-sha1-modp1024	This tells Strongswan to propose aes256 for encryption, sha1 for hashing, and DH group 2 for IKE.
esp	aes256-sha1	This tells Strongswan to propose aes256 for encryption and sha1 for hashing
dpdaction	clear	This means that when a Dead Peer is detected, the VPN will be closed
dpddelay	300s	Interval for DPD packets (seconds)

conn DigiWR		
left	%any	That means that Strongswan accepts Ike connections on any local interfaces using any of it's locally configured IP's
leftcert	SrvCert.pem	Defines the Certificates file to use to authenticate the server itself
leftsubnet	10.0.1.0/24	Server subnet LAN
right	%any	That means that Strongswan accepts Ike connections coming from any IP address
righttsubnet	10.0.0.0/24	Client subnet LAN
Auto	add	That means that, when Strongswan first starts, it should accept this connection, but not try to initiate it.

3.4.2 ipsec.secrets

Edit the ipsec.secrets file using a text editor (nano is used in this example):

```
# This file holds shared secrets or RSA private keys for authentication.

# RSA private key for this host, authenticating it to any other host
# which knows the public part.

: RSA SrvKey.pem
```

In this file, the private key to use for Server needs to be defined.

Please note that syntax is very important here, be sure to have the **":"** with a **space** before the "RSA SrvKey.pem" part, otherwise this will not be recongnized by StrongSwan.

3.4.3 Start the StrongSwan IPsec daemon

In order to have the changes to take effect, the StrongSwan IPsec daemon needs to be restarted using the following command:

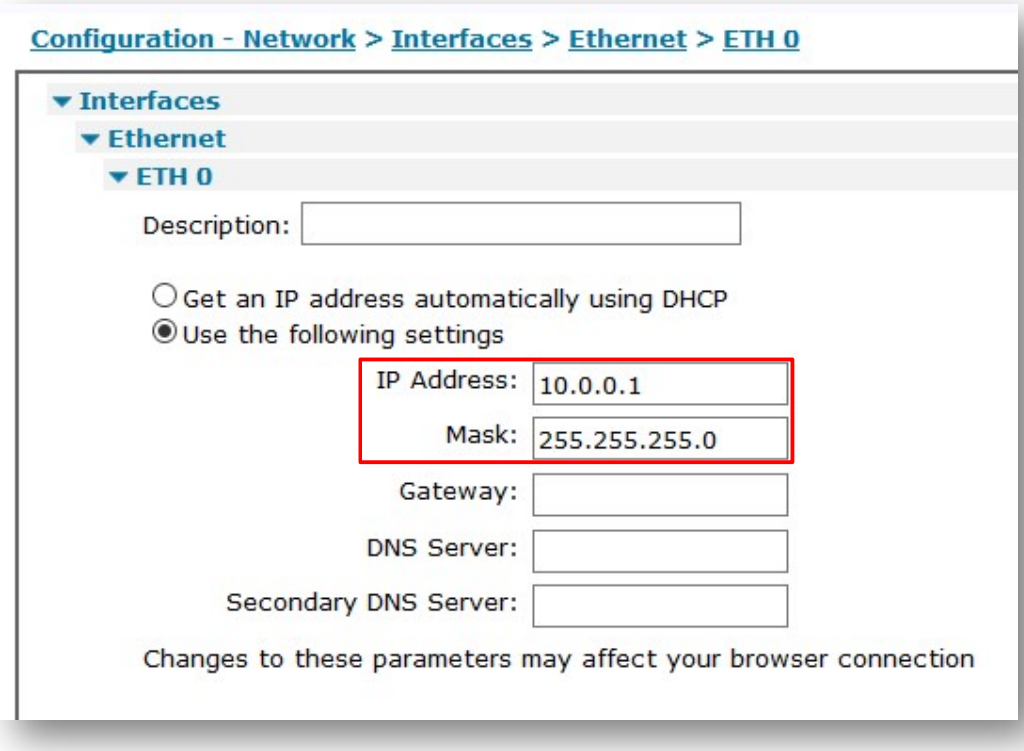
```
digi@Digi:~$ sudo ipsec restart
[sudo] password for digi:
Stopping strongSwan IPsec...
Starting strongSwan 5.5.1 IPsec [starter]...
```

4 TRANSPORT WR CONFIGURATION

4.1 LAN Interface

In this example, the LAN interface is configured with a static address as follows:

CONFIGURATION - NETWORK > INTERFACES > ETHERNET > ETH 0



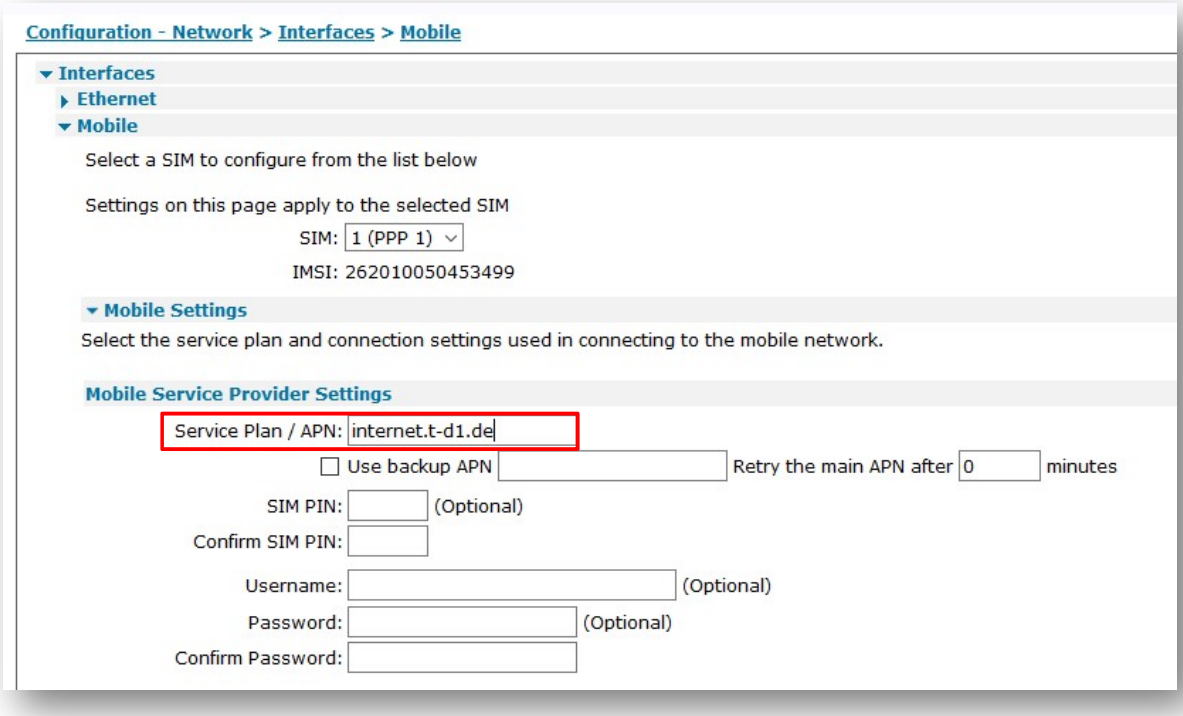
Where:

Parameter	Setting	Description
Use the following settings	Checked	A static IP will be used as defined below
IP Address	10.0.0.1	Enter the IP address of the LAN interface for the router
Mask	255.255.255.0	Enter the subnet mask

4.2 WAN Interface

In this example the WR Router has the Mobile interface as the WAN interface and it is configured as follows:

CONFIGURATION - NETWORK > INTERFACES > MOBILE



Where:

Parameter	Setting	Description
Service Plan/APN	internet.t-d1.de	Enter the APN of your mobile provider

Please note: Depending on provider, a SIM PIN or Username/Password may be required. If needed, enter them in the appropriate fields.

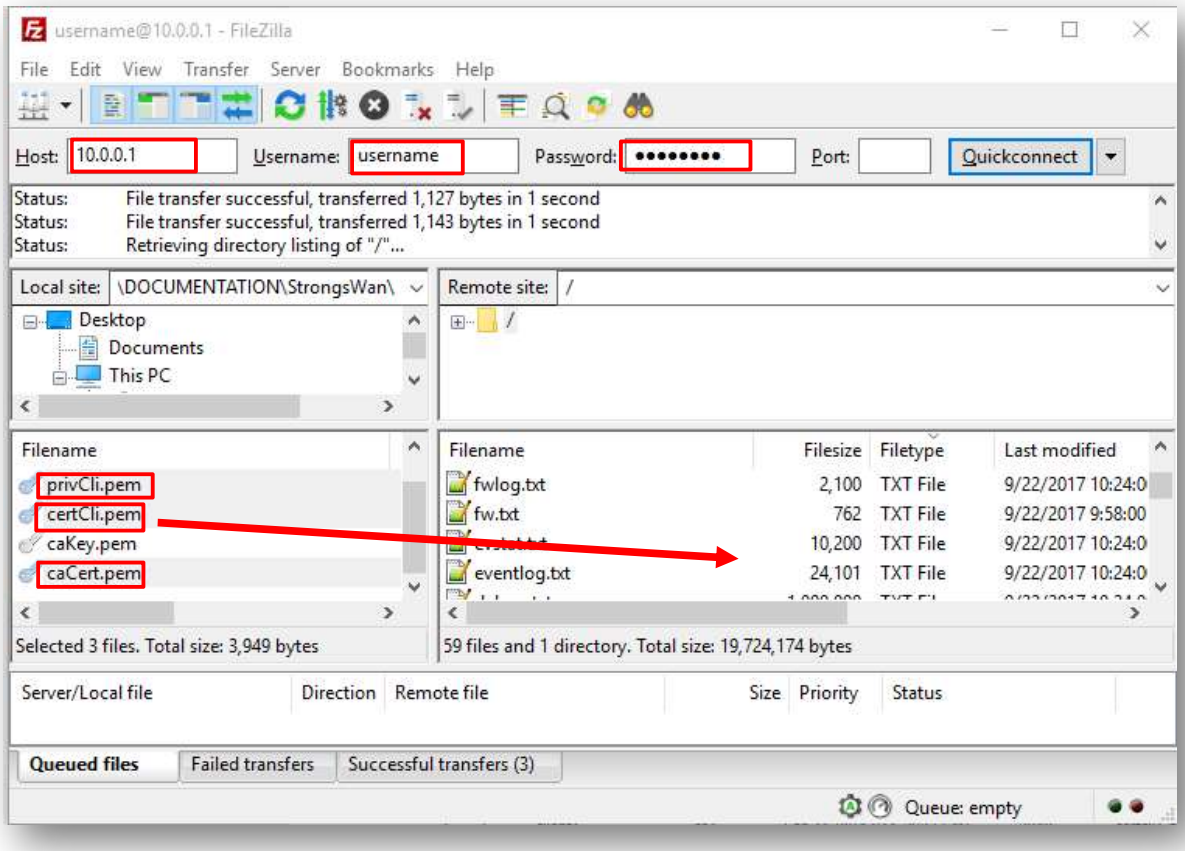
4.3 Uploading Certificates and Keys

Before to transfer the Certficates and Key files on the client, they must be renamed as follows:

Filename	Purpose	New FileName
caCert.pem	Root CA certificate	caCert.pem (*unchanged in this case)
ClientCert.pem	Client Certificate	certCli.pem
CliKey.pem	Client Key	privCli.pem

Once done that, the files can be transferred to the Client using for example an FTP client, connected with the TransPort router with usual username and password.

Please note that you may need to change your IP on the laptop accordingly with the new IP address configured on the ETH0 of the router.



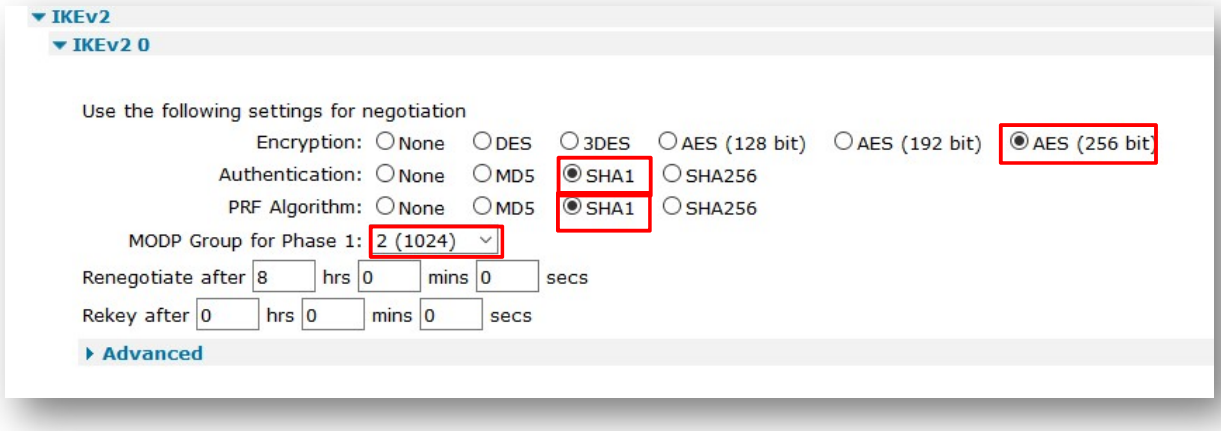
In this example, in order to upload the files, the connection to the Transport is done on the local LAN (so using the ETH o IP address of the router).

4.4 IKEv2/IPsec Tunnel Configuration

Phase 1 and 2 of the IKEv2/IPsec tunnel is configured as follows.

4.4.1 Phase 1 Settings

CONFIGURATION – NETWORK > VIRTUAL PRIVATE NETWORKING (VPN) > IPSEC > IKEv2 > IKEv2 0 and > ADVANCED



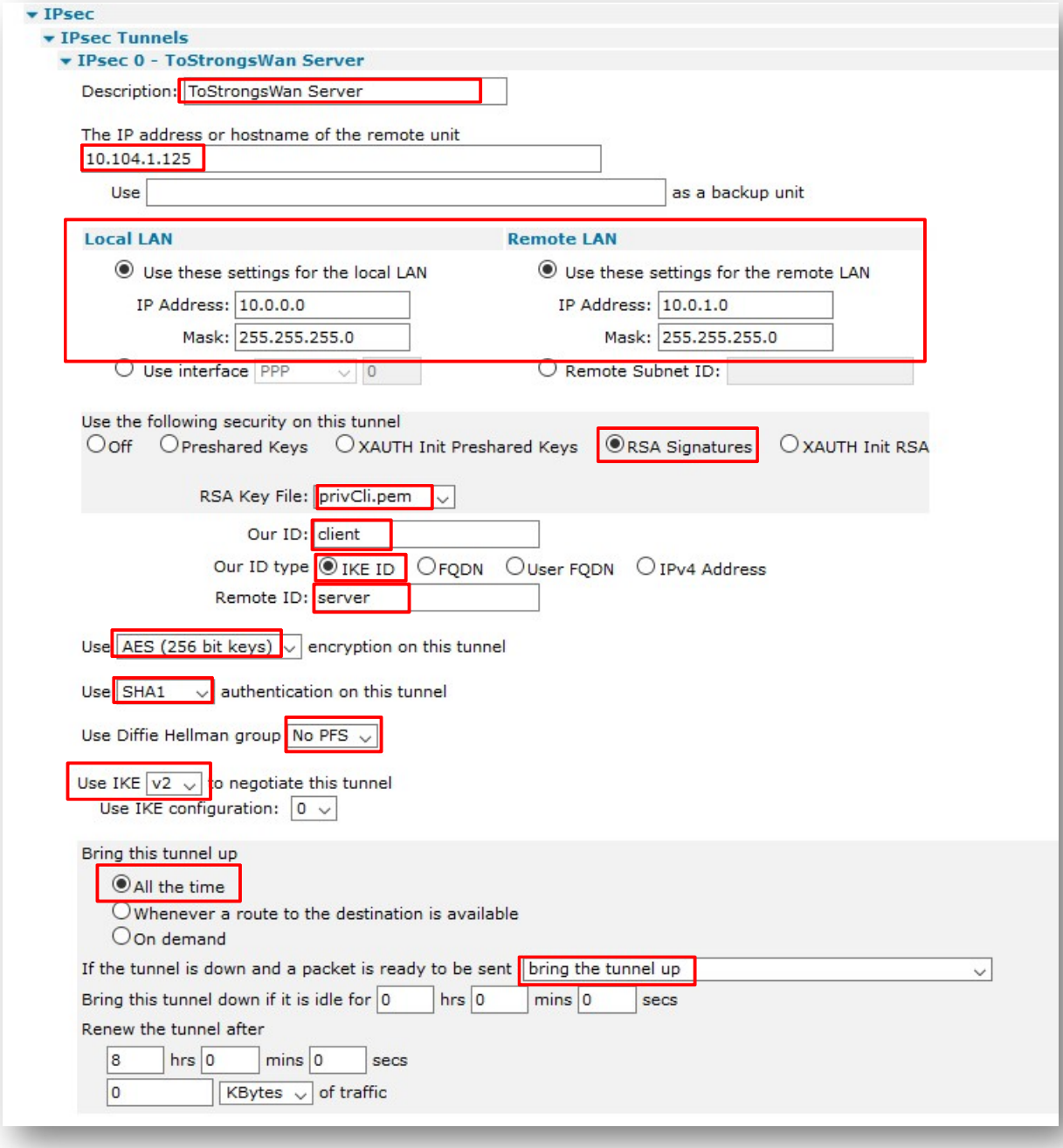
Where:

Parameter	Setting	Description
Encryption	AES (256 bit)	The encryption algorithm used
Authentication	SHA1	The authentication algorithm used
PRF Algorithm	SHA1	The PRF (Pseudo Random Function) algorithm used
MODP Group for Phase 1	2 (1024)	Sets the key length used in the IKE Diffie-Hellman exchange to 768 bits (group 1) or 1024 bits (group 2). In this example group 2 is chosen to enable a 1024 bit key length

4.4.2 Phase 2 Settings

Then, the IPsec tunnel must be configured with the following settings:

CONFIGURATION – NETWORK > VIRTUAL PRIVATE NETWORKING (VPN) > IPSEC > IPSEC TUNNELS > IPSEC 0-9 > IPSEC 0



Where:

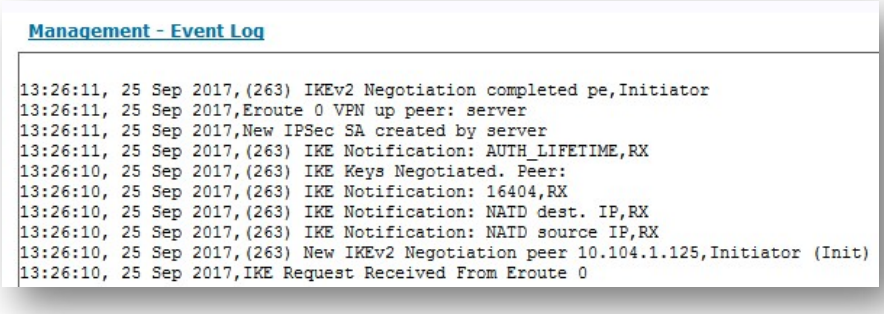
Parameter	Setting	Description
Description	ToStrongSwan Server	Description of the IPsec tunnel
The IP address or hostname of the remote unit	10.104.1.125	The IP address or hostname of the remote IPsec peer that a VPN will be initiated to.
Local LAN IP Address	10.0.0.0	Use this IP address for the local LAN subnet. This is usually the IP address of the router's Ethernet interface or that of a specific device on the local subnet
Local LAN Mask	255.255.255.0	Use this IP mask for the local LAN subnet. The mask sets the range of IP addresses that will be allowed to use the IPsec tunnel.
Remote LAN IP Address	10.0.1.0	Use this IP address for the remote LAN subnet. This is usually the IP address of the peer's Ethernet interface or that of a specific device on the local subnet
Remote LAN Mask	255.255.255.0	Use this IP mask for the remote LAN subnet. The mask sets the range of IP addresses that will be allowed to use the IPsec tunnel.
Use the following security on this tunnel	**RSA Signatures**	Select RSA signature security for this tunnel to use the uploaded certificates
RSA Key File	**privCli.pem**	Private key file used for the responder
Our ID	**client**	ID that is matching the **CN** of the certificate in the first router (**client**)
Our ID type	IKE ID	Defines how the remote peer is to process the Our ID configuration. Set to IKE ID to match the information used in the certificate
Remote ID	**server**	Remote ID that is matching the **CN** in the second router certificate (**server**)
Use ( ) encryption on this tunnel	AES (256 bit keys)	The ESP encryption protocol to use with this IPsec tunnel
Use ( ) Authentication on this tunnel	SHA1	The ESP authentication algorithm to use with this IPsec tunnel
Use Diffie Hellman group ( )	No PFS	The Diffie Hellman (DH) group to use when negotiating new IPsec SAs.
Use IKE n to negotiate this tunnel	**v2**	The IKE version to use to negotiate this IPsec tunnel.
Use IKE configuration	0	The IKE configuration instance to use with this Eroute when the router is configured as an Initiator
Bring this tunnel up	All the time	This controls how the IPsec tunnel is brought up, for the initiator "All the time" option is chosen
If this tunnel is down and a packet is ready to be sent	Bring the tunnel up	Defines the action that is performed when the IPsec tunnel is down and a packet needs to be sent. For the initiator in this AN the "bring the tunnel up" option is chosen

5 CHECK IKEV2/IPSEC TUNNEL STATUS AND TEST

5.1 Check the IPsec tunnel is UP on TransPort WR

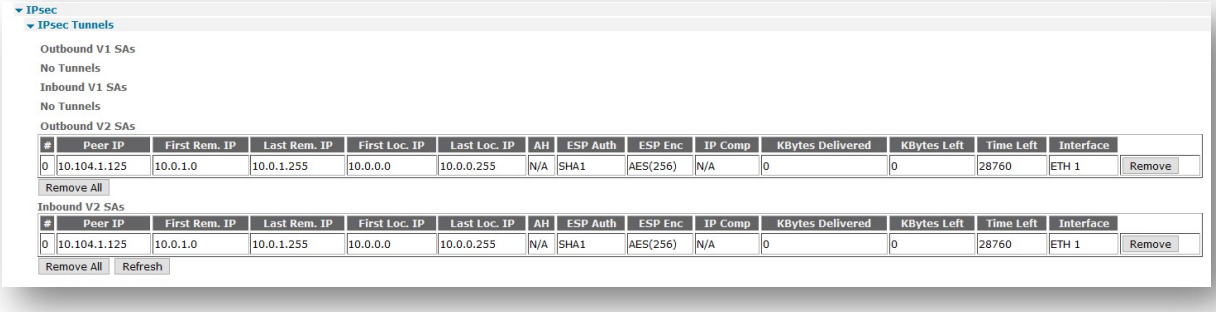
On TransPort WR the Event log will show the IKEv2 negotiation start and ends successfully:

MANAGEMENT – EVENT LOG



After that, in the connections status section IPsec and IKE v2 SAs will be displayed:

MANAGEMENT - CONNECTIONS > VIRTUAL PRIVATE NETWORKING (VPN) > IPSEC > IPSEC TUNNELS



This can be checked also via CLI:

```
sastat
IPsec SAs (total:1). Eroute 0 -> 4
Outbound V1 SAs
        List Empty
Inbound V1 SAs
        List Empty
Outbound V2 SAs
      SPI Eroute         Peer IP                First Rem. IP                  Last
Rem. IP                First Loc. IP                Last Loc. IP    TTL      KBytes
Left                VIP
 c8552c16      0    10.104.1.125                    10.0.1.0
10.0.1.255                     10.0.0.0                 10.0.0.255  28713
0                N/A
Inbound V2 SAs
      SPI Eroute         Peer IP                First Rem. IP                  Last
Rem. IP                First Loc. IP                Last Loc. IP    TTL      KBytes
Left                VIP
 5d0df7eb      0    10.104.1.125                    10.0.1.0
10.0.1.255                     10.0.0.0                 10.0.0.255  28713
0                N/A
OK
```

5.2 Check the IPsec tunnel is UP on StrongSwan

On The StrongSwan server, the IPsec tunnel status can be checked with the command "*sudo ipsec statusall":*

```
digi@Digi:~$ sudo ipsec statusall
[sudo] password for digi:
Status of IKE charon daemon (strongSwan 5.5.1, Linux 4.10.0-35-generic, x86_64):
  uptime: 2 hours, since Sep 25 12:35:49 2017
  malloc: sbrk 1486848, mmap 0, used 403040, free 1083808
  worker threads: 11 of 16 idle, 5/0/0/0 working, job queue: 0/0/0/0, scheduled: 5
  loaded plugins: charon test-vectors aes rc2 sha2 sha1 md4 md5 random nonce x509
revocation constraints pubkey pkcs1 pkcs7 pkcs8 pkcs12 pgp dnskey sshkey pem openssl
fips-prf gmp agent xcbc hmac ccm gcm attr kernel-netlink resolve socket-default
connmark stroke updown eap-mschapv2 xauth-generic
Listening IP addresses:
  10.104.1.125
  10.0.1.100
  10.8.0.1
Connections:
      DigiWR:  %any...%any  IKEv2, dpddelay=300s
      DigiWR:   local:  [C=NL, O=Digi, CN=server] uses public key authentication
      DigiWR:    cert:  "C=NL, O=Digi, CN=server"
      DigiWR:   remote: uses public key authentication
      DigiWR:   child:  10.0.1.0/24 === 10.0.0.0/24 TUNNEL, dpdaction=clear
Security Associations (1 up, 0 connecting):
      DigiWR[8]:    ESTABLISHED    4    minutes    ago,    10.104.1.125[C=NL,    O=Digi,
CN=server]...10.104.1.115[C=NL, O=Digi, CN=client]
      DigiWR[8]:  IKEv2  SPIs:  00000107fffffef8_i  743eedad58684761_r*,  public  key
reauthentication in 2 hours
      DigiWR[8]: IKE proposal: AES_CBC_256/HMAC_SHA1_96/PRF_HMAC_SHA1/MODP_1024
      DigiWR{5}:  INSTALLED, TUNNEL, reqid 2, ESP SPIs: c8552c16_i 5d0df7eb_o
      DigiWR{5}:  AES_CBC_256/HMAC_SHA1_96, 0 bytes_i, 0 bytes_o, rekeying in 39 minutes
      DigiWR{5}:   10.0.1.0/24 === 10.0.0.0/24
```

5.3 Testing

An easy way to test if the LAN to LAN traffic pass through the tunnel, is to generate a ping from each side of the tunnel to reach the remote end's local interface.

From TransPort WR:

From StrongSwan side:

```
digi@Digi:~$ ping 10.0.0.1
PING 10.0.0.1 (10.0.0.1) 56(84) bytes of data.
64 bytes from 10.0.0.1: icmp_seq=1 ttl=250 time=1.47 ms
64 bytes from 10.0.0.1: icmp_seq=2 ttl=250 time=1.48 ms
64 bytes from 10.0.0.1: icmp_seq=3 ttl=250 time=1.23 ms
```

6 CONFIGURATION FILES

6.1 TransPort WR configuration file

The TransPort WR configuration file used for the purpose of this Application Note is shown below:

```
config c show
eth 0 IPaddr "10.0.0.1"
lapb 0 ans OFF
lapb 0 tinact 120
lapb 1 tinact 120
lapb 3 dtemode 0
lapb 4 dtemode 0
lapb 5 dtemode 0
lapb 6 dtemode 0
ip 0 cidr ON
def_route 0 ll_ent "PPP"
def_route 0 ll_add 1
eroute 0 descr "ToStrongsWan Server"
eroute 0 peerip "10.104.1.125 "
eroute 0 peerid "server"
eroute 0 ourid "client"
eroute 0 locip "10.0.0.0"
eroute 0 locmsk "255.255.255.0"
eroute 0 remip "10.0.1.0"
eroute 0 remmsk "255.255.255.0"
eroute 0 ESPauth "SHA1"
eroute 0 ESPenc "AES"
eroute 0 authmeth "RSA"
eroute 0 nosa "TRY"
eroute 0 autosa 2
eroute 0 ikever 2
eroute 0 enckeybits 256
eroute 0 privkey "privCli.pem"
dhcp 0 respdelms 500
dhcp 0 mask "255.255.255.0"
dhcp 0 gateway "192.168.1.1"
dhcp 0 DNS "192.168.1.1"
sntp 0 server "time.devicecloud.com"
dyndns 0 ifent "default"
ppp 0 timeout 300
ppp 1 name "W-WAN (LTE)"
ppp 1 phonenum "*98*1#"
ppp 1 IPaddr "0.0.0.0"
ppp 1 timeout 0
ppp 1 ipsec 1
ppp 1 use_modem 1
ppp 1 aodion 1
ppp 1 autoassert 1
ppp 1 r_chap OFF
ppp 3 defpak 16
ppp 4 defpak 16
web 0 prelogin_info ON
web 0 showgswiz ON
ike 0 deblevel 4
ike2 0 iencalg "AES"
ike2 0 ienckeybits 256
ike2 0 idhgroup 2
modemcc 0 info_asy_add 4
```

```
modemcc 0 init_str "+CGQREQ=1"
modemcc 0 init_str1 "+CGQMIN=1"
modemcc 0 apn "Your.APN.goes.here"
modemcc 0 link_retries 10
modemcc 0 stat_retries 30
modemcc 0 sms_interval 1
modemcc 0 sms_access 1
modemcc 0 sms_concat 0
modemcc 0 init_str_2 "+CGQREQ=1"
modemcc 0 init_str1_2 "+CGQMIN=1"
modemcc 0 apn_2 "Your.APN.goes.here"
modemcc 0 link_retries_2 10
modemcc 0 stat_retries_2 30
modemcc 0 sms_access_2 1
modemcc 0 sms_concat_2 0
ana 0 anon ON
ana 0 l2on OFF
ana 0 l3on OFF
ana 0 xoton OFF
ana 0 lapdon 0
ana 0 lapbon 0
ana 0 maxdata 1500
ana 0 logsize 180
cmd 0 unitid "ss%s>"
cmd 0 cmdnua "99"
cmd 0 hostname "digi.router"
cmd 0 asyled_mode 2
cmd 0 tremto 1200
cmd 0 rcihttp ON
user 0 access 0
user 1 name "username"
user 1 epassword "KD5lSVJDVVg="
user 1 access 0
user 2 access 0
user 3 access 0
user 4 access 0
user 5 access 0
user 6 access 0
user 7 access 0
user 8 access 0
user 9 access 0
local 0 transaccess 2
sslcli 0 verify 10
sslsvr 0 certfile "cert01.pem"
sslsvr 0 keyfile "privrsa.pem"
ssh 0 hostkey1 "privSSH.pem"
ssh 0 nb_listen 5
ssh 0 v1 OFF
templog 0 mo_autooff ON
cloud 0 ssl ON

Power Up Profile: 0
OK
```

6.2 StrongSwan Server configuration files

The StrongSwan configuration files used for the purpose of this Application Note are shown below:

ipsec.conf

```
# ipsec.conf - strongSwan IPsec configuration file
# basic configuration

config setup

# Add connections here.


conn %default
    keyexchange=ikev2
    ike=aes256-sha1-modp1024
    esp=aes256-sha1
    dpdaction=clear
    dpddelay=300s

conn DigiWR
    left=%any
    leftcert=SrvCert.pem
    leftsubnet=10.0.1.0/24
    right=%any
    rightsubnet=10.0.0.0/24
    auto=add
```

ipsec.secrets

```
# This file holds shared secrets or RSA private keys for authentication.

# RSA private key for this host, authenticating it to any other host
# which knows the public part.

: RSA SrvKey.pem
```