IPsec

bintec Dm739-I

Legal Notice

Warranty

This publication is subject to change.

bintec offers no warranty whatsoever for information contained in this manual.

bintec is not liable for any direct, indirect, collateral, consequential or any other damage connected to the delivery, supply or use of this manual.

Table of Contents

I Related Documents

bintec Dm719-I IP Tunnel

bintec Dm752-I Access Control

bintec Dm754-I NSLA

bintec Dm759-I VRRP Protocol

bintec Dm790-I LDAP Protocol

Chapter 1 Introduction

1.1 Virtual private networks

Until now, businesses have traditionally used the Internet to promote their products and services through websites. Now, an increasing number of companies use the Internet to connect their different offices, branches or production and development centers. In short, the Internet can displace costly and inflexible private lines. In addition, e-business needs the global access (World Wide Web) offered by the Internet.

Packets circulating on public networks, like the Internet, travel through multiple nodes that we cannot control or monitor. Packets to the same destination can take different paths, therefore we need to establish security mechanisms that keep intruders from accessing the data we send over this type of network.

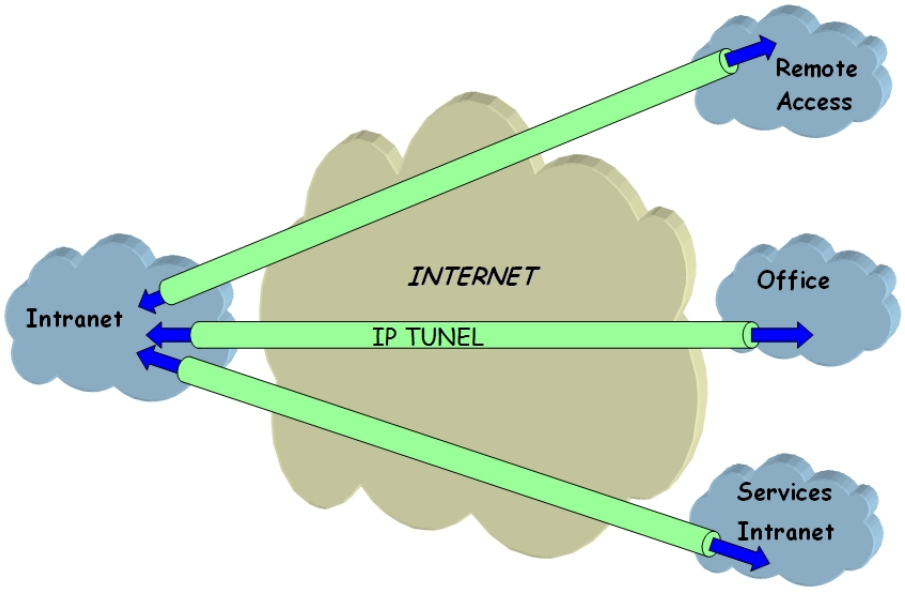


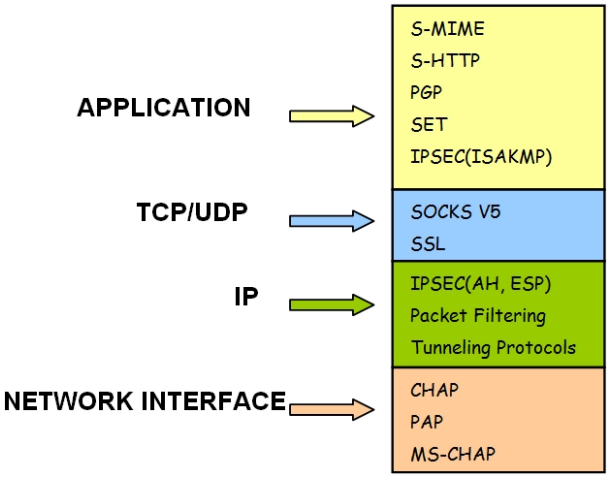
Fig. 1: Virtual Private Network

A virtual private network (VPN) aims to extend a company's Intranet across a public network such as the Internet by creating a secure connection with private tunnels.

There are different types of VPN solutions which may be classified according to the OSI layer where they are implemented:

- VPNs implemented at the *Application* layer : These authenticate and/or encrypt the message but not the source and destination address of the packets they route.
- *Link* layer-based VPNs: Like L2TP, these can only authenticate the tunnel endpoints, not each individual packet.
- VPNs implemented at the *Network* layer: Like IPsec, these protect the data and the source and destination addresses without the user having to modify any applications, but offer no protection outside the tunnel (e.g., in the company's Intranet).

To conclude, a combination of application and network layer VPNs is advisable to ensure adequate security.



1.2 IPsec

IPsec is a network layer security platform developed by the IPsec Working Group of the Internet Engineering Task Force (IETF). It provides a flexible and robust way to accommodate new encryption and authentication algorithms.

IPsec focuses on the following security issues:

- Data source authentication: verifying that the received data has been sent by those claiming to have sent it.
- Data integrity: verifying that the received data has not been altered during transmission.

  The term data authentication is often used to indicate both data integrity and source authentication.
- Data confidentiality: concealing the data using an encryption algorithm.
- Anti-replay protection: preventing hackers from retransmitting messages without the sender's knowledge.
- Automated cryptographic key management.

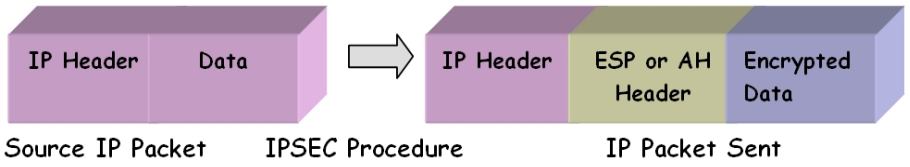
IPsec has defined two different security services to address these issues:

- **Encapsulating Security Payload** (ESP): This provides confidentiality, source authentication, integrity and protection against replay attacks.
- **Authentication Header** (AH): This provides source authentication, integrity and protection against replay attacks, but it does not offer confidentiality. This service should be used when you need to ensure the origin of the data.

1.2.1 IPsec tunnels

The IPsec platform allows two modes of operation, and either security service (AH or ESP) may be used in each:

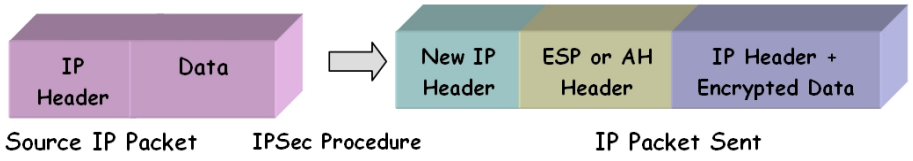
- **Transport mode:** allows secure communication, normally established between two hosts (e.g., communication between a workstation and a server or between two servers) but never masks the source and destination address of the packet. In transport mode, IPsec only acts on the internal data of the IP packet, without altering the header (e.g., on a TCP or UDP segment or an ICMP packet).



Note

In Transport mode, if there are several clients behind the same NAT, the TCP or UDP protocol must be configured in the access lists and, even then, only one client will be able to open each service simultaneously against the same server (e.g., 1 Telnet connection or 1 SSH connection). For further details, please see section 5.2 "Transport Mode Conflict" of RFC 3948 "UDP Encapsulation of IPsec ESP Packets".

- **Tunnel mode:** the entire original IP packet is encapsulated in a new IP packet, thereby concealing all of the original content. In this way, information can travel through a tunnel from one point in the network to another without anyone examining it. This mode is the best method to use for communications between a router and an external host, or between two routers.



1.2.2 IPsec architecture

1.2.2.1 Security policy database (SPD)

The IPsec platform needs to know which security policy to apply to the IP packet, based on the header fields (also called *selectors*). Security policies decide which encryption and authentication algorithms to use in the secure connection.

The Security Policy Database (SPD) stores entries containing control selectors and their associated security policies.

After carrying out a search of applicable actions for a packet in the security policy database, there are three possibil-

ities:

- Discard the packet.
- Route the packet normally.
- Apply IPsec with certain encryption and authentication algorithms. These algorithms will depend on the security-performance compromise adopted. For example, if processing speed is more important than security, you should choose DES encryption rather than Triple DES.

1.2.2.2 Security associations (SA)

When a packet's selectors match an SPD entry, the packet is processed according to the policy associated with that entry. A security association (SA) is the secure connection created after consulting the SPD and contains the security information (encryption and authentication keys) necessary to process a packet.

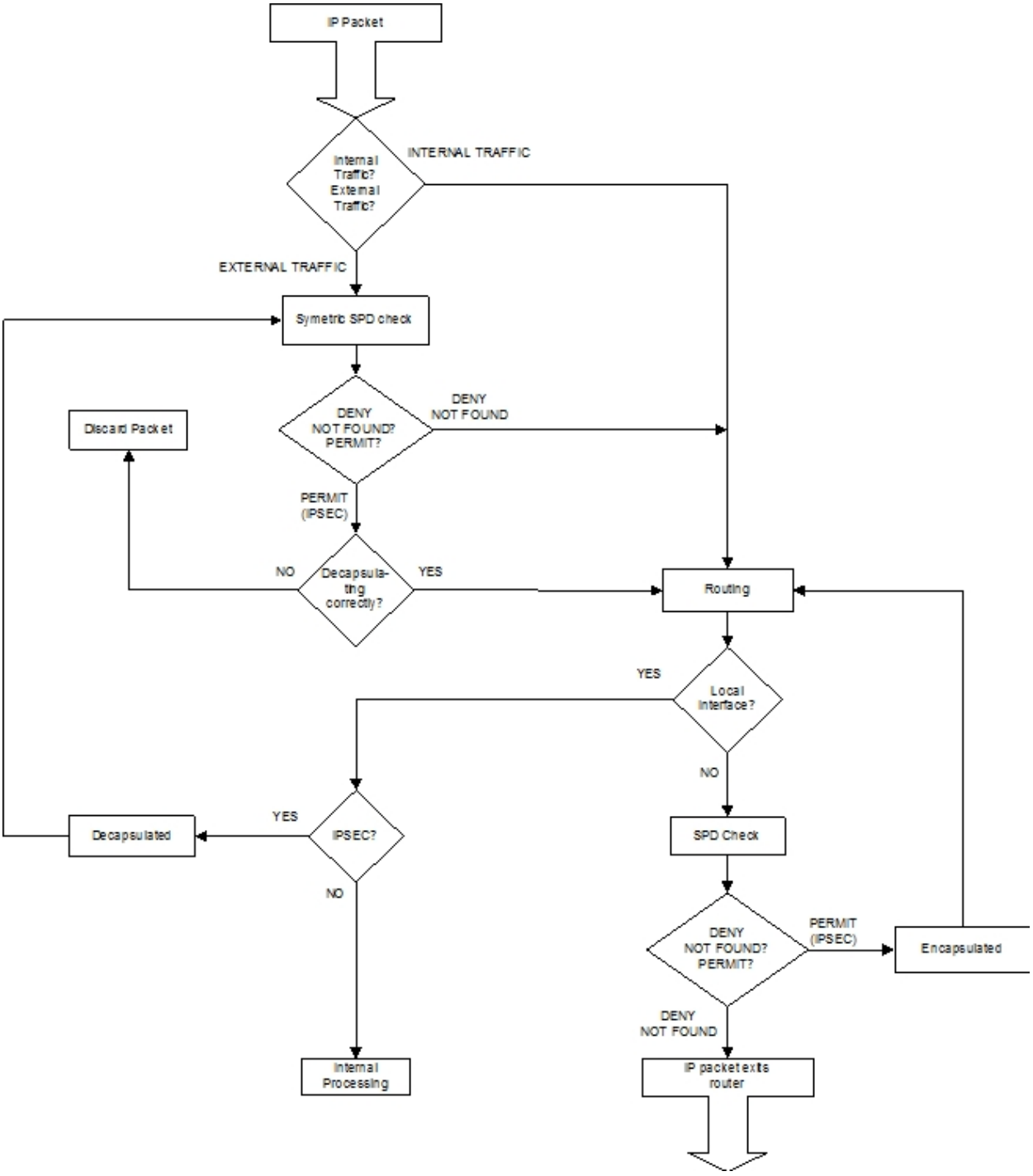
Depending on the security service (ESP or AH) specified, you can choose between several different cryptographic (DES, TRIPLE DES, etc.) and authentication (MD5, SHA1, etc.) algorithms.

1.2.2.3 IPsec packet processing

The SPD or policy database is defined by the user and is used for traffic going out of the router. Inbound traffic is controlled by means of a symmetrical *implicit* SDP. This way, all incoming packets are processed according to the provisions set for outbound packets: if outbound traffic is to to be sent with a particular IPsec security policy, then the matching inbound traffic should also comply with the same policy. Similarly, if the policy dictates routing/dropping the outbound traffic, the inbound traffic would also be routed/dropped.

After the internal routing is done, the SPD is consulted again (this time to determine whether to drop, route or apply IPsec processing to outbound traffic).

The following flowchart shows how an IP packet proceeds when the IPsec protocol is invoked on our routers.



1.2.3 Advanced IPsec

1.2.3.1 Key management

All secret key-based security platforms stop being secure if the keys are not renewed regularly.

A shorter refresh interval provides our system with greater protection against cryptanalysis.

Generally speaking, there are two ways to manage IPsec keys and security parameters: manual (manual IPsec) and automatic or dynamic (IPsec IKE). These modes refer to the way in which peers reach agreement on tunnel security parameters.

1.2.3.2 Manual IPsec

In manual IPsec (manual-keying), the keys used in the encryption and/or authentication process for each Security Association (SA) are entered by the user. The same parameters (keys, encryption and authentication algorithms) must be set at both tunnel endpoints to allow the secure communication to go ahead. This is practical in small, relatively static environments. However, when the VPN starts to grow, the practice of renewing the keys manually may become too costly a task.

1.2.3.3 IKE IPsec

Thanks to the IKE Internet Key Exchange protocol (based on OAKLEY key exchange and ISAKMP), the key renewal process can now be fully automated. The two tunnel endpoints automatically negotiate the parameters of the secure communication (keys, encryption and authentication algorithms). Before this negotiation can take place, the tunnel endpoints must first negotiate phase I security parameters to protect the exchange. The first phase is also when the tunnel endpoints authenticate one another. They do this using a common pre-shared key manually introduced at both ends, digital certificates or a public key algorithm.

There are two pre-negotiation modes: *main mode* and *aggressive mode*.

- *Main mode* masks the identities of the tunnel endpoint routers. In this type of negotiation, both endpoints need to know the IP addresses of the security server at the other end of the tunnel.
- *Aggressive mode* does not mask these identities and speeds up the authentication process. It also allows the negotiation to go ahead without having to know the IP address of the server at the other end of the tunnel. Thus, a tunnel can be set up with an unknown security router if the security policy associated with the packet allows this.

IPSec IKE has four phase I operating modes, depending on the type of authentication used to negotiate the Security Association's security parameters.

1.2.3.3.1 Pre-shared key authentication

The two security routers are able to mutually authenticate by using a common pre-shared key that is manually configured on both devices.

There are two types of exchange with pre-shared key authentication: *main mode* and *aggressive mode*:

- *Main mode* masks the identities of the tunnel endpoint routers.
- *Aggressive mode* does not mask these identities and speeds up the authentication process.

Whenever a Security Association's (SA) lifetime expires, new keying material will be exchanged between the two security routers after authentication with the manual pre-shared key.

With IPsec manual keying and IPsec pre-shared key authentication, the tunnel endpoint IP address (i.e., the security router's IP address) must be known.

However, the following IPsec IKEs allow tunnels to be set up automatically and dynamically with unknown security routers (if the packet's security policy allows this). With these types of IPsec IKE, you are not required to configure a private key on the tunnel endpoints because it is automatically obtained through the processes described below.

1.2.3.3.2 Digital signature authentication

The two-tunnel endpoints are authenticated by means of a digital signature and the Diffie-Hellman key exchange scheme.

There are two types of exchange: *main mode* and *aggressive mode*:

- *Main mode* masks the identities of the tunnel endpoint routers.
- *Aggressive mode* does not mask these identities but speeds up the authentication process.

1.2.3.3.3 Public key encryption authentication

The RSA public key cryptosystem is used for authentication when the other router's public key is known. Certificates may be used to obtain the public keys of the other tunnel endpoint.

There are two types of exchange: *main mode* and *aggressive mode*. If the public key is frequently updated, the former of these two modes is just as secure and faster than the latter.

Furthermore, *Public Key Encryption Authentication* provides more security vis-à-vis *Digital Signatures* and *Pre-shared Key Authentication* by combining the RSA public key and Diffie-Hellman key exchange systems. On the downside, processing time is much longer.

1.2.3.3.4 Revised public key encryption authentication

RSA is used for authentication when the other router's public key is known. Certificates may be used to obtain the public keys of the other end of the tunnel.

Although public key operations are reduced (with a negligible loss of security), authentication services are improved.

Two types of exchange exist: *main mode* and *aggressive mode*. If the public key is frequently updated, the former of these two modes is just as secure and faster than the latter.

1.2.3.4 IKEv2

Although IKEv2 fulfills the same purpose as IKEv1 (as described above), it simplifies the SA negotiation process by only using two pairs of IKE messages, a request and a response (called an exchange), whereas IKEv1 uses at least 6.

The first pair, IKE SA INIT, negotiate cryptographic algorithms, exchange nonces, and execute a Diffie-Hellman exchange.

The second pair of messages, IKE AUTH, authenticate the previous messages, exchange identities and certificates, and establish the first Child SA.

The IKEv2 protocol, like IKEv1, is used to negotiate ESP or AH SAs.

Every IKE message contains a Message ID as part of its fixed header. Said Message ID is used to match up requests and responses, and to identify message retransmissions.

IKEv2 messages contain the following payloads:

Notation	**Payload**
AUTH	Authentication.
CERT	Certificate.
CERTREQ	Certificate Request.
CP	Configuration.
D	Delete.
EAP	Extensible Authentication.
HDR	IKE header (not a payload).
IDi	Identification - Initiator.
IDr	Identification - Responder.
KE	Key Exchange.
Ni, Nr	Nonce.
N	Notify.
SA	Security Association.
SK	Encrypted and Authenticated.
TSi	Traffic Selector - Initiator.
TSr	Traffic Selector - Responder.
V	Vendor ID.

The initial exchanges are as follows:

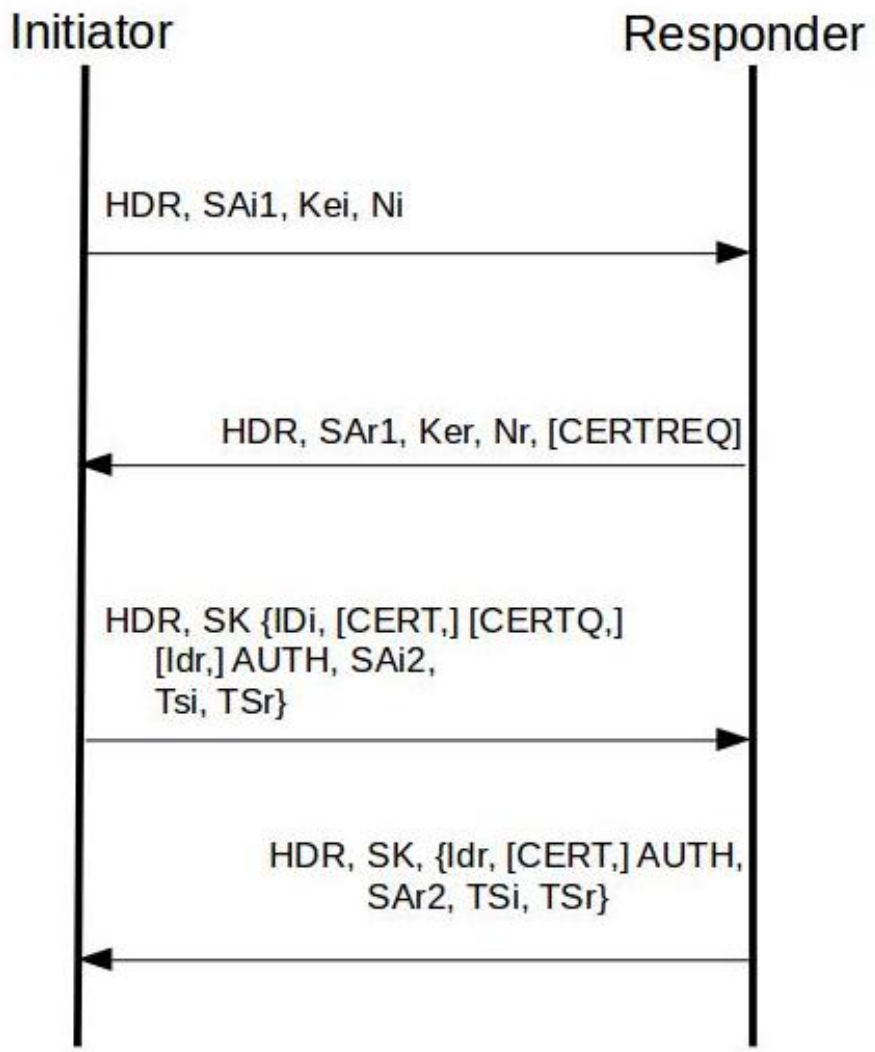


Fig. 2: Initial exchanges

For further information on IKEv2 operations, please see RFC 7296.

1.2.3.5 High security

The keys used to encrypt or authenticate a connection are obtained from keying material. If this material has not and will not be used to generate any additional keys to encrypt or authenticate other connections, then we can say that perfect forward secrecy (PFS) has been achieved.

Our routers allow PFS when they are configured to run in high security mode, but this comes at the cost of a higher computational cost in setting up the IPsec tunnels.

High security mode also generates more secure keying material using Oakley Groups more resilient to cryptanalysis.

1.2.3.6 Certificates

Certificates allow routers to the learn the public keys of other security routers with which to open an IPsec tunnel. These public keys will be used in the two IKE public key authentication modes.

1.2.3.7 Tunnel end-point discovery (TED)

The TED protocol is an IPsec enhancement that allows a security router to dynamically determine the end router that should be used to set up an IPsec tunnel in order to guarantee a secure communication between the hosts protected by the two routers.

To configure a large, fully-meshed network, you need to define static security parameters for all possible pairs in the network. By using TED and a single set of dynamic security parameters, you can find the pair you are looking for without it being predefined. New links can also be added to the network without having to modify the configuration of every router residing in the network.

When the TED protocol is used, the hosts protected by the routers must have routable IP addresses. In addition, these addresses will be sent without encryption; therefore, the protocol should not be used in scenarios where this information is considered confidential. You must also ensure that the corresponding access list contains only IP-related entries (and therefore cannot be used with UDP, TCP or any other protocol).

Note

Protected IP addresses must be routable addresses.

1.2.3.8 Reverse route injection (RRI)

The reverse route injection (RRI) algorithm allows an IPsec tunnel endpoint router to insert static routes to networks protected by this tunnel in its corresponding routing tables. These routes are inserted when the IPsec tunnel is opened, and indicate how to reach the network (with mask) protected by the access list associated with the tunnel, with a next hop router defined by configuration (this next hop can be the local or remote tunnel endpoint, or a user-defined IP address).

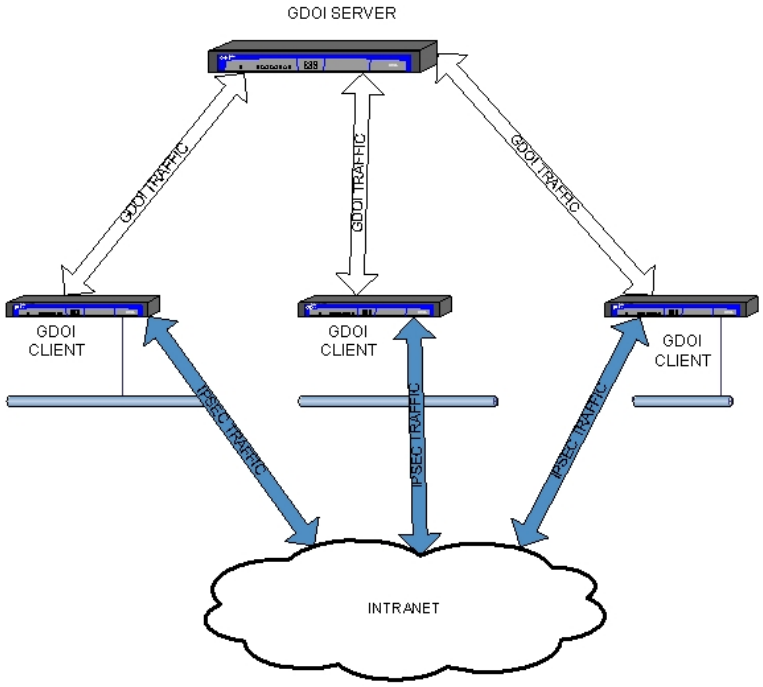
The ultimate aim of this functionality is to propagate these routes backwards by means of a routing algorithm (for example, RIP or OSPF), thus allowing the devices behind the router to learn the path needed to send the encrypted traffic to the network(s) protected by the tunnel.

1.2.3.9 Group Domain Of Interpretation (GDOI)

Definition:

Group Domain Of Interpretation (GDOI) is defined as the Internet Security Association Key Management Protocol (ISAKMP) Domain of Interpretation (DOI) for group key management. The GDOI protocol operates between a client or group member (GM) and a central server or key server (GCKS). The central server establishes security associations (SAs) between authorized group members. The ISAKMP protocol described in RFC 2048 defines two negotiation phases; the GDOI protocol is protected by ISAKMP phase I, while phase II is completely different and is defined in RFC 3547.

The diagram below shows the two different types of traffic involved: GDOI traffic between clients and server, and IPSEC traffic between clients using the policies downloaded through the GDOI protocol.



Operative:

A client registers with the server to get the Security Associations (SA) needed to communicate with other clients in the group. During the negotiation, the client sends a group ID to the server and the server sends back the policies and keys for that particular group. Before the current keys expire, they are periodically refreshed through rekey messages to ensure that traffic is not lost. The server is responsible for maintaining and updating the keys and IPsec policies.

The server can send two different encryption keys: the key encryption key (KEK) and the traffic encryption key (TEK). TEK keys are used to encrypt IPsec packets exchanged between clients, while KEK keys are used to encrypt rekey messages exchanged between server and client.

The server sends rekey messages when the TEK or KEK keys need updating and when the server configuration is changed. The server can be configured to retransmit the rekey messages a set number of times to avoid losing the rekey packets. Rekey messages can be sent through unicast IP packets addressed to each client registered with the server, or through a packet addressed to a configurable multicast IP.

When the server sends rekey messages through unicast IP packets addressed to each client registered with the server, such clients may send a rekey acknowledgement to the server (rekey-ack message).

Note

RFC 8263 "GDOI GROUPKEY-PUSH ACK Message" is supported on GDOI Client as of version 11.01.06.

IPsec encapsulation:

Packets encapsulated by GDOI clients are encapsulated in transport or tunnel mode, depending on what the server says. If nothing is heard from the server, GDOI clients encapsulate packets in the configured mode.

Client access list:

The GDOI client receives the access list from the server telling it what traffic will be sent encrypted and what traffic will be sent in clear. The entries on this list must match at least one of the entries on the access list configured in the GDOI client. Each entry on the received access list is installed in front of its matching entry (the permit/deny field is not visible).

There are two basic configurations:

- Configure a **permit** all entry:

```
access-list x
   entry 1 default
   entry 1 permit
exit
```

In this case,

- Before connecting to the server:

  - If unencrypted traffic is received, it is dropped.

  - No traffic is sent until the server is connected.

- After connecting to the server:

  - The downloaded access list entries installed in front of entry 1 determine whether traffic is encrypted.

- Configuring a **deny** all entry:

```
access-list x
   entry 1 default
   entry 1 deny
exit
```

In this case,

- Before connecting to the server:

  - Any unencrypted traffic received is admitted.

  - The traffic is sent in clear.

- After connecting to the server:

  - The downloaded access list entries installed in front of entry 1 determine whether traffic is encrypted.

In addition, exceptions to the ones indicated by the server can be added by adding entries to the access list associated with the GDOI template.

For example, if you associate the following list with a GDOI template in a client, the traffic between hosts 172.24.1.1 and 172.24.1.2 will be sent in clear (unless the server has explicitly specified that the traffic must be sent encrypted):

```
access-list x
   entry 1 default
   entry 1 deny
```

```
         entry 1 source address 172.24.1.1 255.255.255.255
         entry 1 destination address 172.24.1.2 255.255.255.255
;
         entry 2 default
         entry 2 permit
;
      exit
```

If the list in the following example is associated with a GDOI template in a client, then the traffic between hosts 172.24.1.1 and 172.24.1.2 will be sent encrypted (unless the server has explicitly specified that the traffic must be sent in clear):

```
      access-list x
         entry 1 default
         entry 1 permit
         entry 1 source address 172.24.1.1 255.255.255.255
         entry 1 destination address 172.24.1.2 255.255.255.255
;
         entry 2 default
         entry 2 deny
;
      exit
```

Time-based anti-replay protection:

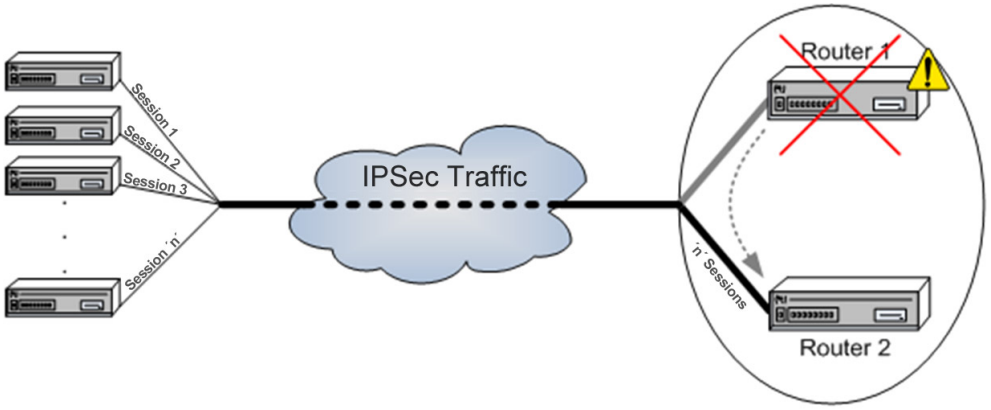
Anti-replay is an important IPsec feature that prevents third parties from eavesdropping packets from an IPsec conversation and playing them back later as if they had been generated at that time. Time stamp-based anti-replay ensures that illegally reproduced packets are detected and discarded. Our implementation of GDOI uses a synchronous anti-replay (SAR) mechanism that is independent of the date and time of the devices. Clients synchronize with a global timestamp sent by the server (GCKS), and update it as the seconds elapse. When a client sends a packet, a proprietary format timestamp is added to the IP packet. This is compared with the current timestamp in the client receiving the packet. If the received timestamp and the current one differ by more than a given configurable value, the packet is dropped.

1.2.3.10 Fault-tolerant IPsec recovery

Fault-tolerant IPsec recovery is a feature that allows our devices to continue managing IPsec packets even when one of our tunnel endpoint devices has stopped working.

1.2.3.10.1 First steps

Fault tolerant IPsec recovery is based on the dynamic distribution of IPsec sessions between a pair of routers, i.e., sessions can be moved from one device to another depending on the current condition and configuration of the devices. In this way, if one of the devices stops working, its IPsec sessions can be automatically and transparently established in the device that is still active.



Fault-tolerant IPsec recovery is supported in VRRP and IPsecFT, as well as in IPsec:

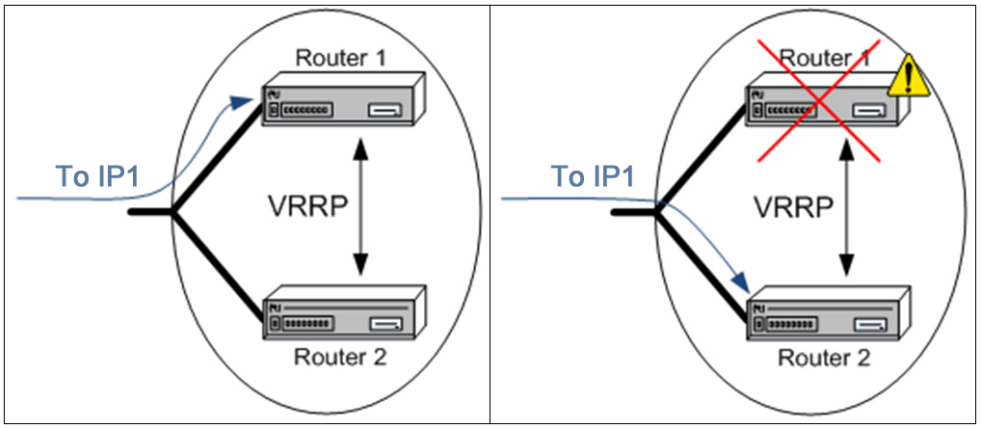
- Virtual Router Redundancy Protocol (VRRP) is defined in RFC 3768 and is responsible for dynamically assigning the virtual router function to one of the VRRP routers. This protocol decides which device should route the packets addressed to the IP address shared by VRRP and, therefore, serves as a basis when deciding what IPsec sessions should be established on which router at any point.
- IP Security Fault Tolerant (IPsecFT) is the protocol responsible for exchanging information between the two devices acting as tunnel endpoints. Through IPsecFT, the two devices maintain an updated database containing sufficient information to enable either of them to inherit, at any point, the IPsec sessions established by their partner.

Both protocols run alongside each other to ensure the sessions corresponding to IPsec are established at all times.

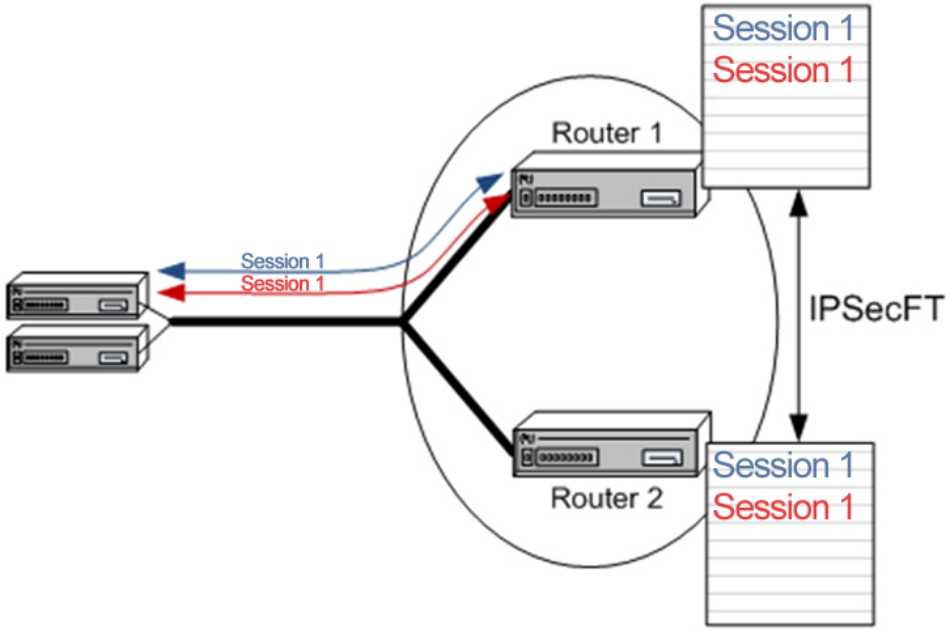
1.2.3.10.2 Operation

Let's take a closer look at how the subsystem works. The two routers are given the same virtual IP addresses to make them look like one device to the outside. This way, when external devices start IPsec sessions, they do so with these shared addresses, regardless of which device is in control at the time.

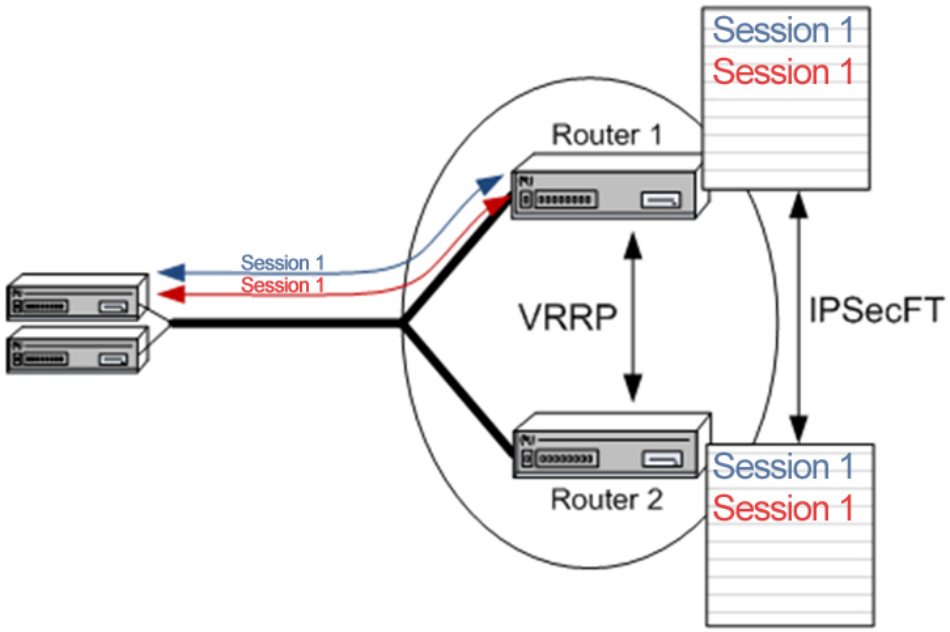
VRRP ensures that one of the devices is always associated with the virtual IP address. For further information on this, please see *bintec* manual *Dm759-I VRRP Protocol*.



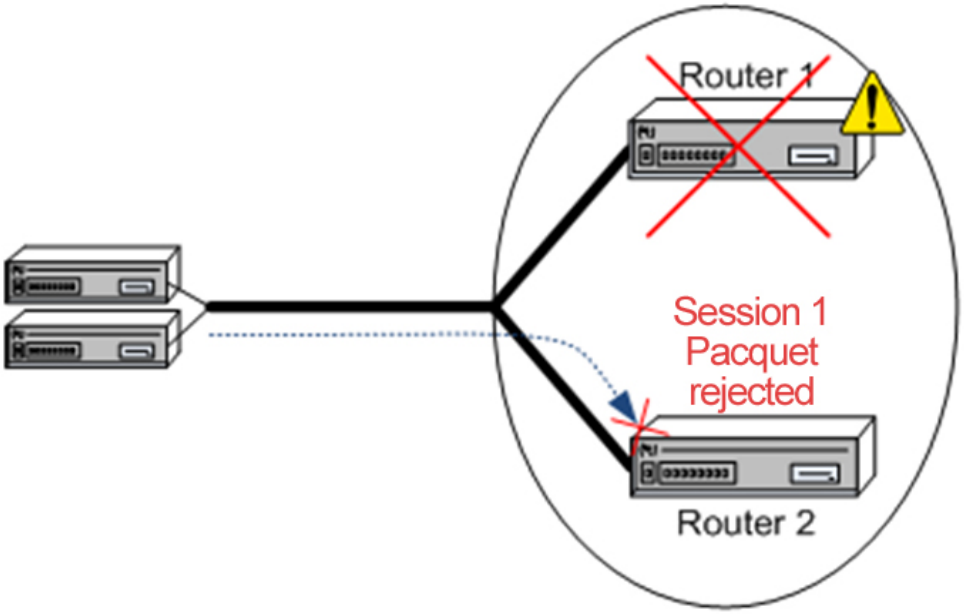
Furthermore, an IPsec session requires keys to be established and exchanged in order for the device to understand and accept the packets sent. In other words, it isn't enough just to solve the problem of managing the IP addresses used to connect the IPsec sessions, we also need to ensure the continuity of those IPsec sessions. The IPsecFT protocol is used for this. As mentioned earlier, this protocol keeps a database with information about the sessions established on the analogue device, that is, it could establish them if necessary.



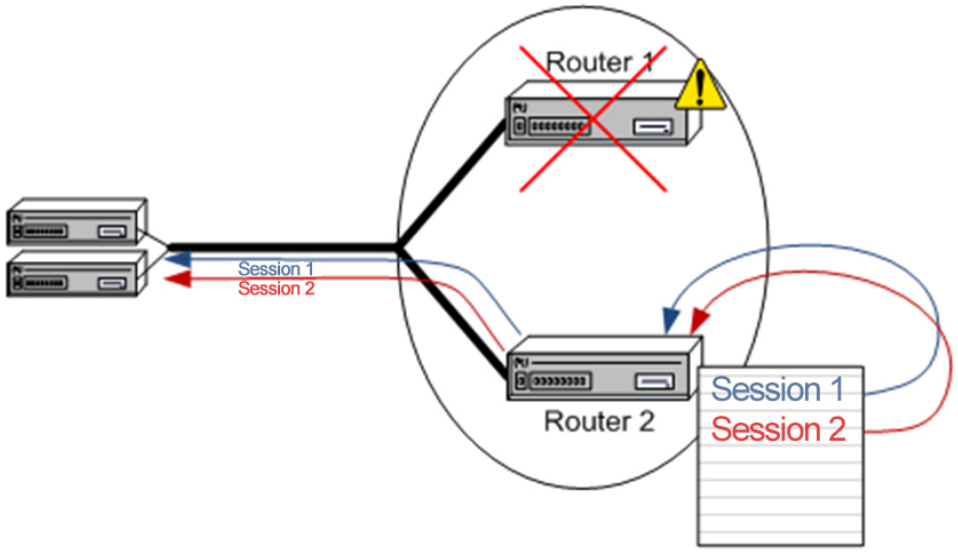
In a stable situation, sessions are established with the device managing the IP address of said sessions. This way, IPSec encrypted data is received and decrypted without difficulty.



Imagine the case that the device with the established sessions fails for some reason and is unable to continue managing said IPsec sessions – for example, it shuts down. The shared IP address will then be managed by the router that is still up. This router would therefore start receiving IPsec packets that it doesn't understand (the IPsec session was not established with it).



However, the device that is still active doesn't wait for IPsec packets that it can't understand. As soon as it sees that it is managing the shared virtual IP address, it searches the IPsecFT session database and establishes any sessions that it doesn't have established if it sees that it will be receiving traffic from those sessions.



Thus, one device can assume responsibility for the IPsec sessions established in another device should it fail, and the changeover is automatic and transparent.

1.2.3.10.3 Important operation considerations

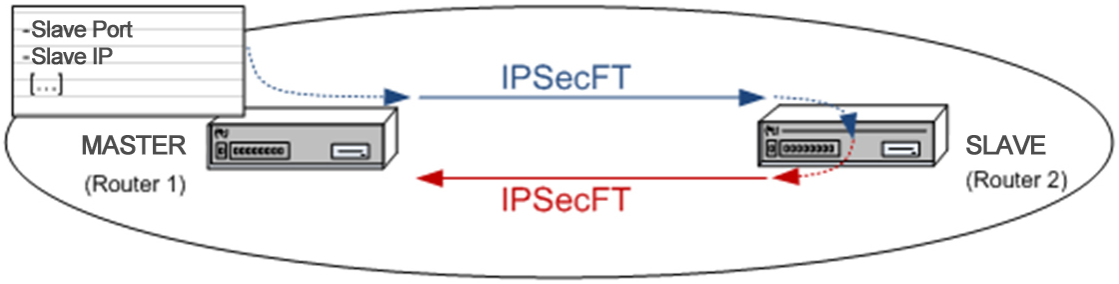
There are certain parameters in setting fault-tolerant IPsec recovery that deserve special mention. As already stated, this function is reliant upon the IPsec, IPsecFT and VRRP protocols.

IPsec

The IPsec configuration controls the procedure for setting up the IPsec sessions. In other words, it configures the keys, encryption type, and any other parameters typical of conventional IPsec configuration. Given that the two devices will be establishing the same sessions, it follows that they will need to have the same IPsec configuration.

IPsecFT

IPsecFT establishes and maintains two TCP sessions for exchanging information relating to the IPsec sessions established in each router. It uses a TCP session in each direction: from Router1 to Router2 and from Router2 to Router1. However, for the sake of end-user convenience, one router is designated master and the other slave. Only the master knows which slave to connect to: the slave device that receives the connection automatically establishes the second session in the opposite direction.



Note

For the protocol to work correctly, it doesn't matter which device is configured as master and which is configured as slave, and this in no way interferes with anything related to master and slave in VRRP.

TCP sessions progress within a set time frame once established. That is to say, IPsecFT completes all outstanding tasks within a preset time period. The two most important tasks are:

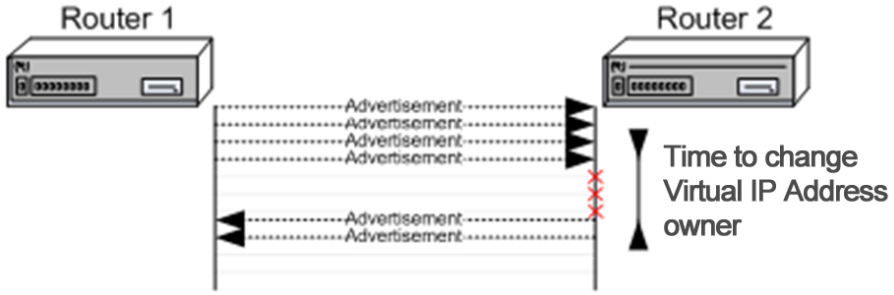
- Sending keep alive packets to monitor the IPsecFT session. If no keep alive packet is received within a preset time, then the session is invalidated and a minute later released.
- Polling the shared virtual IP address. If the address belongs to us, IPsec sessions are established.

If a session is lost, IPsecFT attempts to reestablish it (at one-second intervals).

VRRP

VRRP uses the multicast advertisements, sent by the controlling device at regular intervals, to decide which device controls the shared virtual IP address. The listening device takes control of the virtual IP address when it doesn't re-

ceive advertisement packets for a certain period of time. Once it takes control of the shared address, it starts establishing the IPsec sessions necessary to continue data transmission. The less time the device waits before taking control of the shared IP address, the less time the system will take to recover when the controlling device fails; too little time, however, can make it look as though an active device is down because it can't serve the advertisement packets quickly enough.



General considerations

When the IPsec sessions are sent from one device to another, the receiving device is subject to a high workload. Special care must be taken to select an appropriate number of input buffers on the interface where the sessions are established, otherwise the receiving device will not be able to process all of the incoming packets from the remote devices.

Likewise, you should also increase the IPsec encryption queue size to (at least) the number of simultaneous sessions to be established, or up to the number of input buffers on the interface that establishes the sessions, whichever is greater.

1.2.3.11 IPsec tunnel protection

IPsec tunnel protection provides a routable interface for terminating IPsec tunnels and simplifies the configuration process to protect the remote links.

This makes it possible to provide protection in an IP tunnel after configuring the interface. For further information, please see *bintec* manual *Dm719-IP Tunnel* .

Since there is a routable interface at the tunnel endpoint, many common interface capabilities can be applied to the IPsec tunnel; it allows flexibility for sending and receiving both unicast and multicast IP encrypted traffic (as it supports dynamic routing protocols such as RIP).

Traffic is encrypted or decrypted when forwarded from or to TNIP and is managed by the IP routing table. Using IP routing to forward traffic to TNIP (encrypted) simplifies the IPsec configuration because access lists are not required.

The IPsec source and destination are the same as for the IP tunnel interface (TNIP).

Chapter 2 Configuration

2.1 Introduction

As we saw in section *IPsec architecture* on page 3, IPsec processes an IP packet by applying the security policy configured for that packet. This information is stored in the security policy database (SPD) where the associated selectors and security policies are found. Therefore, configuring IPsec in a router basically comes down to defining the SPD elements.

There are three steps to configuring an SPD element in our router. First, you define an access control list entry (ACE), that is to say, certain control selectors. You do this by assigning a preconfigured generic access list to IPsec. Each entry in the list is configured with an action. This action can be either to apply (permit) the processing corresponding to the protocol assigned to the list, which in this case is IPsec, or not to apply (deny) the corresponding processing. If none of the entries in the list are applicable, the packet will not be processed by IPsec. Then you create the templates or IPsec security policies; this is where the IPsec tunnel security parameters are defined. And finally, an access control list assigned to IPsec is associated or mapped to a specific template.

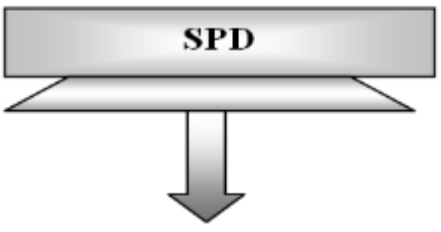
Access 1 control list	
Entry 1	✓ Source IP ✓ Permit ✓ Protocol
Entry 2	✓ Source IP ✓ Permit ✓ Ports ✓ Connection
...	...
Entry n	✓ Source IP ✓ Deny ✓ Protocols

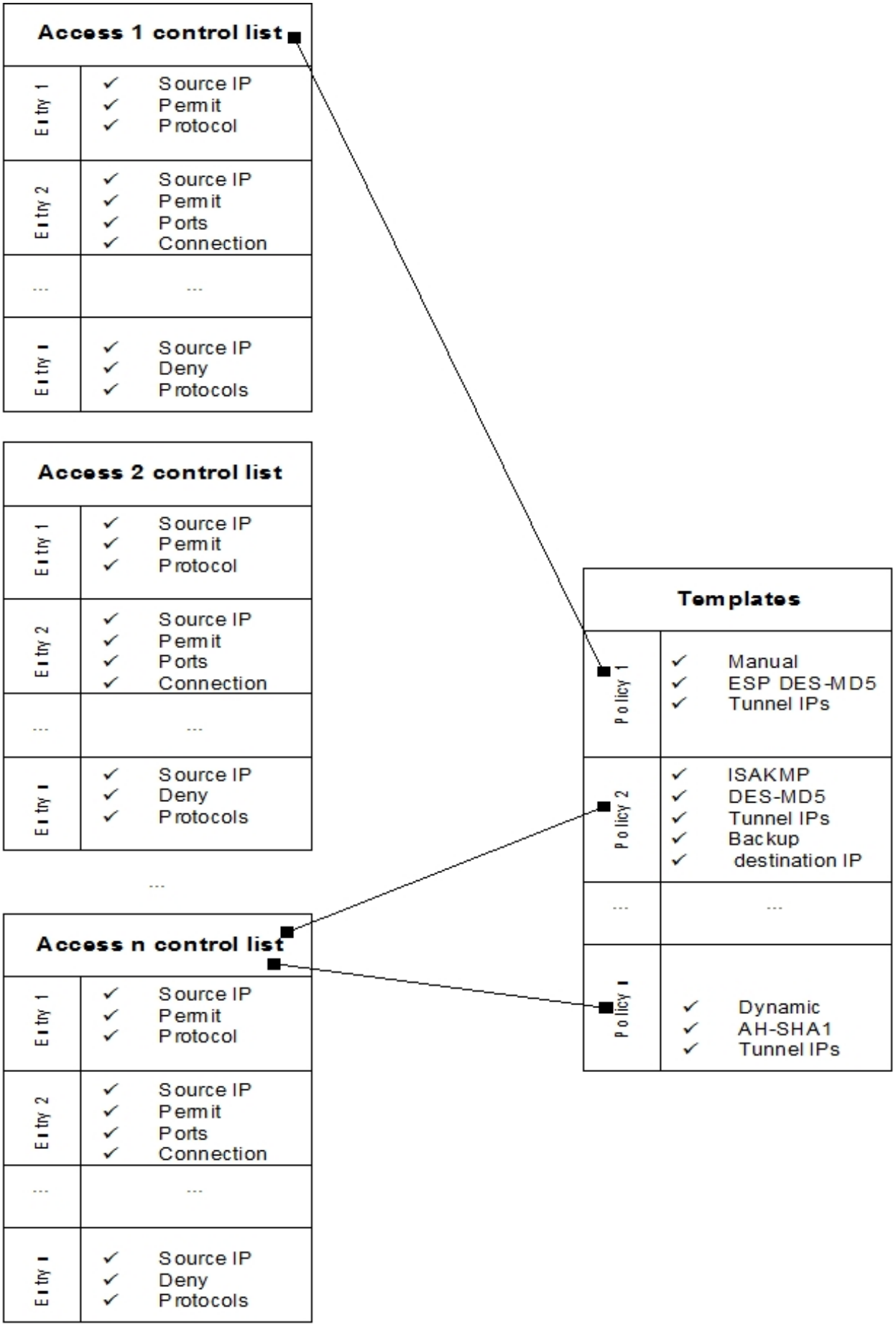
Access 2 control list	
Entry 1	✓ Source IP ✓ Permit ✓ Protocol
Entry 2	✓ Source IP ✓ Permit ✓ Ports ✓ Connection
...	...
Entry n	✓ Source IP ✓ Deny ✓ Protocols

...

Access n control list	
Entry 1	✓ Source IP ✓ Permit ✓ Protocol
Entry 2	✓ Source IP ✓ Permit ✓ Ports ✓ Connection
...	...
Entry n	✓ Source IP ✓ Deny ✓ Protocols

Templates	
Policy 1	✓ Manual ✓ ESP DES-MD5 ✓ Tunnel IPs
Policy 2	✓ ISAKMP ✓ DES-MD5 ✓ Tunnel IPs ✓ Backup ✓ destination IP
...	...
Policy n	✓ Dynamic ✓ AH-SHA1 ✓ Tunnel IPs




Access 1 control list
Entry 1
✓ Source IP
✓ Permit
✓ Protocol
Entry 2
✓ Source IP
✓ Permit
✓ Ports
✓ Connection
...
...
Entry n
✓ Source IP
✓ Deny
✓ Protocols
Access 2 control list
Entry 1
✓ Source IP
✓ Permit
✓ Protocol
Entry 2
✓ Source IP
✓ Permit
✓ Ports
✓ Connection
...
...
Entry n
✓ Source IP
✓ Deny
✓ Protocols
...
Access n control list
Entry 1
✓ Source IP
✓ Permit
✓ Protocol
Entry 2
✓ Source IP
✓ Permit
✓ Ports
✓ Connection
...
...
Entry n
✓ Source IP
✓ Deny
✓ Protocols
Templates
Policy 1
✓ Manual
✓ ESP DES-MD5
✓ Tunnel IPs
Policy 2
✓ ISAKMP
✓ DES-MD5
✓ Tunnel IPs
✓ Backup
✓ destination IP
...
...
Policy n
✓ Dynamic
✓ AH-SHA1
✓ Tunnel IPs

2.2 First steps

2.2.1 Initial configurations

Since anyone with access to the device can modify the IPsec parameters, the first thing to do is to configure the access passwords for Telnet and the device console.

Make sure that the device is set up with the correct date and time to prevent validation problems when using certificates.

DISABLE/ENABLE commands

IPsec can be disabled using the *DISABLE* command from the IPsec configuration menu.

```
Config>protocol ip
-- Internet protocol user configuration --
IP config>ipsec
-- IPSec user configuration --
IPSec config>disable
IPSec config>
```

Type in the *ENABLE* command to enable it.

2.3 IPsec configuration

2.3.1 Correct order for a good configuration

After connecting the device to both the private and public networks, you should configure the SPD for incoming and outgoing packets.

To ensure that the configuration is correct, we recommend you follow the steps below:

(1) Configure the IPsec access control list.

(2) Configure the templates (security parameters).

(3) Create the SPD.

2.3.2 Configuration

This section describes the steps required to configure IPsec on the router. Enter the following commands to access the IPsec configuration protocol environment:

```
Config>protocol ip
-- Internet protocol user configuration --
IP config>ipsec
-- IPSec user configuration --
IPSec config>
```

The following commands are available within the IPsec configuration environment (indicated by the IPsec config> prompt).

Command	Operation
? (HELP)	Lists the available commands or options.
ENABLE	Allows you to enable IPsec and filter the events to be viewed.
DISABLE	Disables IPsec.
DESCRIPTION	Adds a description to IPsec.
NO	Deletes elements from the templates and access control lists, undoes mappings or deletes the entire configuration.
ASSIGN-ACCESS-LIST	Assigns an access control list to the IPsec protocol.
TEMPLATE	Configures security policy parameters for IPsec tunnels.
MAP-TEMPLATE	Associates (maps) an access control list element with a template.
ASSOCIATE-KEY	Associates a key with an access control list.
ASSOCIATE-DEST-MASK	Associates a destination mask with an access control list.

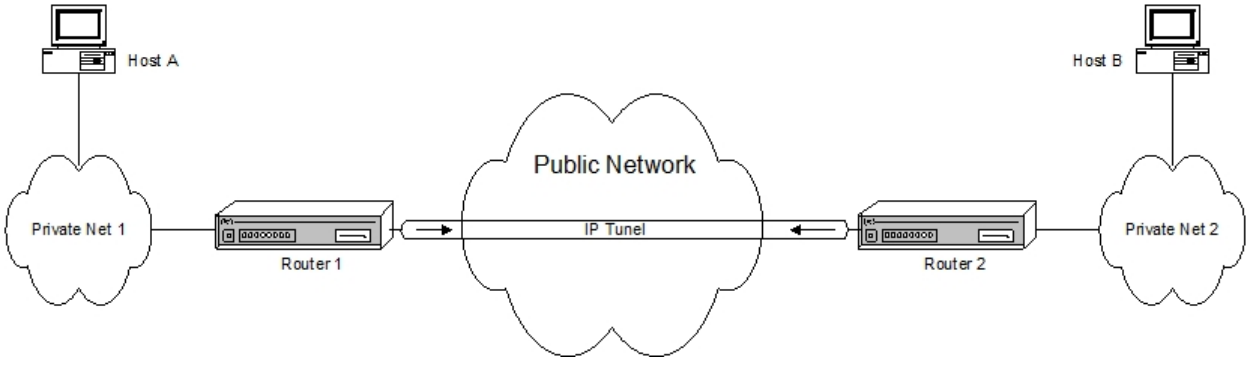
KEY	This is used and described in the section on dynamic templates (IPsec IKE).
EVENT	Allows you to configure a filter that either limits the events shown or displays all of them.
QOS-PRE-CLASSIFY	Enables packet pre-filtering (for BRS).
ADVANCED	Allows you to configure advanced parameters.
LDAP	Allows you to specify operational parameters to perform LDAP queries.
LIST	Lists the IPsec configuration.
CERT	Accesses the digital certificates submenu.
REPORT	Report statistics.
XAUTH-HOSTNAME	Configuration of xauth user associated with hostname.
XAUTH-IP	Configuration of xauth user associated with IP.
XAUTH-USER	Configuration of xauth user properties.
EXIT	Exits the IPsec configuration prompt.

The device will request any required command line parameters that are missing, unless there is an option to write subcommands. In either case, you can get help by entering the name of the command or subcommand followed by a question mark ('?').

```
IPSec config>?
  enable                 Enables IPSec
  disable                Disables IPSec
  description            Add a description to IPSec
  no                     Negate a command or set its defaults
  assign-access-list     Assigns access lists to IPSec (used as SPD selectors)
  template               Configures security policies params for IPSec tunnels
  map-template           Associates an element in the LCA with a template
  associate-key          Associates a key with an access list
  associate-dest-mask    Associates a destination mask with an access list
  key                    Adds preshared or RSA keys
  event                  Adds a filter for IPSec events or enables all of them
  qos-pre-classify       Enables QOS Preclassify
  advanced               Configuration of advanced IPSec parameters
  ldap                   Configures parameters to perform LDAP operations
  list                   Lists the IPSec configuration
  cert                   Enters the Digital certificates submenu
  report                 Report statistics
  xauth-hostname         Configures xauth user associate to hostname
  xauth-ip               Configures xauth user associate to ip
  xauth-user             Configures xauth user properties
  gdoi                   Configures Group Encrypted Transport
  fault-tolerant         Configures IPSec fault tolerant
  exit                   Exits IPSec configuration menu
IPSec config>
```

2.3.2.1 IPsec access control list configuration

As explained earlier, there is an access control list. Each entry in this list consists of a block of selectors and an action, and is identified by a unique number (the entry identifier or ID field). The block of selectors is composed of a source IP address (or range of addresses), a destination IP address (or range of destination IP addresses), a protocol (or range of protocols), source and destination ports (or range of ports), and the connection identifier between the interfaces through which the packet travels. You don't have to specify all of them, just the ones you wish. The action represents the processing assigned to the packets (**permit** or **deny**) matching the associated block of selectors.



As already mentioned when analyzing the SPD, ACL entries or elements are always established for packets going out of the routers. Say we want to set up a secure IPsec tunnel for packets being routed between host A and host B as shown in the above figure. To do this, the ACL control entry must contain, at least, the following selectors:

- Host A *source IP address*;

- Host B *destination IP address*;

- Action: PERMIT (IPsec processing);

Any packet traveling from A to B would thus be encapsulated by IPsec. Implicit in defining this entry is that any packet arriving from host B with address A would have to have the same encapsulation. Thus the secure tunnel is completely defined between both ends.

Note

The order of the entries in the access control list is important when the information referenced by the selectors matches multiple ACL elements.

Note

However, the order is not determined by each entry's identifier but by the order in which the entries are listed (which can be modified). Hence, if the list is searched, starting with the first element or entry on the list, and a match is found, the action specified in the ACL entry is applied and no more entries are checked.

IPsec makes use of the generic and extended access control lists defined in the root menu of the device configuration. For further information, see *bintec Dm752-I Access Control*.

The lists created in this menu must be assigned to the IPsec protocol through the **IPsec config>** *ASSIGN-ACCESS-LIST* command. The order in which these lists are assigned determines the query order applied to the processed packets.

A generic and extended access control list is made up of a series of *entries* which define the properties a packet must have to be considered as belonging to a particular entry, and consequently to that list. The generic access control list is then assigned to a protocol.

Once you have configured the access control list and assigned it to IPsec, you can run the *LIST ACCESS-LISTS-ALL-ENTRIES* command to view all the entries on the access list. Alternatively, you can run the *LIST ACCESS-LISTS ADDRESS-FILTER-ENTRIES* command from the IPsec config> menu to filter the entries that you view by source or destination address.

```
IPSec config>list access-lists all-entries
Extended Access List 100, assigned to IPSec
11    DENY    SRC=192.168.4.8/32  DES=192.168.10.27/32  Conn:0
        PROT=1-9  SPORT=21-25  DPORT=1000-2000

10    PERMIT  SRC=192.168.4.0/24  DES=192.168.10.0/24  Conn:0
        PROT=1-9  SPORT=21-25  DPORT=1000-2000
```

```
IPSec config>list access-lists address-filter-entries 192.168.4.8 255.255.255.255
Extended Access List 100, assigned to IPSec
11    DENY    SRC=192.168.4.8/32  DES=192.168.10.27/32  Conn:0
        PROT=1-9  SPORT=21-25  DPORT=1000-2000
```

2.3.2.2 Configuring templates (security parameters)

Templates are IPsec security policies that can be associated with one or more elements from the access control list (ACL). Only generic lists that have been previously assigned to IPsec may be associated with a template.

Each template is defined with the two tunnel endpoint addresses (corresponding to the security routers at each end of the tunnel), the authentication/encryption algorithms, IPsec key management manual (manual IPsec) or dynamic mode (IKE IPsec), and a template identifier (ID).

Each mode has a series of associated commands, some of which are common to both modes. When you list the commands in the template, you will be shown the meanings of the configured mode, which is why we need to define the template mode before doing anything else:

```
IPSec config>template 1 ?
  manual     Manual template
```

```
isakmp     Isakmp template
dynamic    Dynamic template
ikev2      Ike version 2 template
```

We will now describe manual IPsec before going on to look at IPsec IKE (IKEv1 and IKEv2).

2.3.2.2.1 Manual templates

In a manual IPsec configuration (manual-keying), the user is responsible for entering the keys used to encrypt data and/or authenticate a Security Association (SA). The user must ensure that the same security parameters (keys, authentication and encryption algorithms) are configured at both ends of the tunnel in order for the secure communication to take place.

Manual templates are configured using the following subcommands within the *TEMPLATE* command:

Command	Operation
DEFAULT	Restores a template to its default settings.
MANUAL	Creates a static template with a security service (ESP or AH).
SOURCE-ADDRESS	Adds the tunnel's source address to the template.
DESTINATION-ADDRESS	Adds the tunnel's destination address to the template.
SPI	Adds the security configuration (SA) ID number defined by the template.
KEY	Adds a DES key to the template.
TKEY	Adds a Triple DES key to the template.
MD5KEY	Adds an MD5 key to the template.
SHA1KEY	Adds a SHA1 key to the template.
ANTIREPLAY	Activates anti-replay in the template.
DF-BIT	Specifies how to process the DF bit in IPsec packets.
MTU-THRESHOLD	Specifies the minimum MTU threshold to use in PMTU processing.
MTU-DEFAULT	Specifies the initial MTU value given through the IPsec tunnel.
NO	Negates a command or restores the default value.

The first thing that needs to be defined when configuring a template (manual or dynamic) is the security service to use: Encapsulating Security Payload (ESP) or Authentication Header (AH). ESP provides data confidentiality services by encrypting the data and also has an option to perform authentication. AH only allows authentication:

“TEMPLATE [ID] DEFAULT”

Restores a template back to its default settings.

Example:

```
IPSec config>template 1 default
```

Command history:

Release	Modification
11.00.05	The *TEMPLATE DEFAULT* command is obsolete as of version 11.00.05.
11.01.00	The *TEMPLATE DEFAULT* command is obsolete as of version 11.01.00.

“TEMPLATE [ID] MANUAL ESP [ENCRYPT] [AUTHEN]”

Used to define a manual template with the ESP security service.

The two encryption algorithms that can be selected in a manual template are: Data Encryption Standard (DES) and Triple Data Encryption (TDES). Advanced Encryption Standard (AES) cannot be selected in a manual template.

The following authentication algorithms may be chosen: MD5, SHA1, SHA256, SHA384, SHA512 or NONE.

The ID field is the template's identification number.

Example:

```
IPSec config>template 1 manual esp des md5
```

“TEMPLATE [ID] MANUAL AH [AUTHEN]”

Used to define a manual template with the AH security service.

The following authentication algorithms may be chosen: MD5, SHA1, SHA256, SHA384, or SHA512.

The ID field identifies the template.

Example:

```
IPSec config>template 5 manual ah sha1
```

Having defined the security service, the tunnel endpoint IP addresses, the SA identifier created from the template (SPI), and the keys that will be used with the chosen encryption and authentication algorithms must be entered.

“TEMPLATE [ID] SOURCE-ADDRESS [IP ADD/INTERFACE]”

Adds the tunnel's local IP address to the template identified by [ID]. You can specify the interface that the IP address is taken from.

Example:

```
IPSec config>template 1 source-address 192.100.1.2
```

“TEMPLATE [ID] DESTINATION-ADDRESS [IP ADD/DOMAIN NAME]”

Includes the IP address/domain name of the remote end of the tunnel in the template identified by [ID].

Example:

```
IPSec config>template 1 destination-address 192.100.1.1
```

In this particular case, 192.100.1.1 is used as the IP destination address.

Example:

```
IPSec config>template 1 destination-address XXX.bintec.de
```

In this case, the domain name “XXX.bintec.de” is used as the destination. Keep in mind that you will need to configure a DNS server to resolve the domain name.

“TEMPLATE [ID] SPI [INTEGER > 256]”

Allows you to include the Security Parameter Index (SPI) value for the template identified by [ID]. The SPI is an integer, [INTEGER], greater than 256. The SPI must be the same at both ends and distinguishes one template from another when both have the same tunnel destination address and security service (ESP or AH).

Example:

```
IPSec config>template 1 spi 280
```

To delete the configured SPI value, use the **no** form of this command.

You cannot define two policies with identical values for the three parameters mentioned: Tunnel destination address, security service and SPI.

“TEMPLATE [ID] KEY [8 bytes key]”

This is used to add a key when DES encryption is chosen. The template's DES encryption key is represented by an 8-byte key (specify this number either in hexadecimal format, prefixed with 0x, or ASCII).

Example:

```
IPSec config>template 1 key 0x0123456789ABCDEF
```

To delete the configured KEY, use the **no** form of this command.

Please note, if you enter a hexadecimal key, you must enter double the number of characters (between 0-9 and A-F), since two hexadecimal characters define one byte.

“TEMPLATE [ID] TKEY [24 bytes key]”

This is used when Triple DES is chosen as the encryption algorithm. The Triple DES key length is 24 bytes (specify this number either in hexadecimal format, prefixed with 0x, or ASCII).

Example:

```
IPSec config>template 1 tkey 0123456789abcdefghijklmn
```

To delete the configured TKEY, use the **no** form of this command.

“TEMPLATE [ID] MD5KEY [16 bytes key]”

When MD5 is used for authentication, you must enter a 16-bit key (specify this number either in hexadecimal format, prefixed with 0x, or ASCII).

Example:

```
IPSec config>template 1 md5key bintecsabintecsa
```

Use the **no** form of this command to delete the configured MD5KEY.

“TEMPLATE [ID] SHA1KEY [20 bytes key]”

When SHA1 is used for authentication, you need to enter a 20-byte key (specify this number either in hexadecimal format, prefixed with 0x, or ASCII).

Example:

```
IPSec config>template 1 sha1key bintecsabintecsa1234
```

Use the **no** form of this command to delete the configured SHA1KEY.

“TEMPLATE [ID] ANTIREPLAY”

This command enables anti-replay protection, which protects against replay attacks (retransmission of captured communications). The command is enabled by default but we recommend disabling it on manual templates. Use the **no** form of this command to do this.

Example:

```
IPSec config>template 1 no antireplay
```

“TEMPLATE [ID] DF-BIT {SET/CLEAR/COPY}”

When encapsulating a packet in IPsec, it is the router's job to ensure the Path Maximum Transmission Unit Discovery (PMTUD) algorithm continues to function in hosts protected by the tunnel. The PMTUD algorithm uses the *don't fragment* (DF) bit in the IP header and ICMP packets. This option lets you specify the way in which the router processes the DF bit in IPsec packets: *always mark* (all IPsec traffic leaves with the DF bit set to TRUE), *always remove* (the bit is set to FALSE, ICMP/PMTUD packets are not processed so the router acts as a black hole for this algorithm), or *copy* the packet being protected (normal ICMP/PMTUD processing and router default option). For further information, please see Section 6 of RFC 1191.

Example:

```
IPSec config>template 1 df-bit ?
set     set the DF bit on the IPSec packets
clear   clear the DF bit on the IPSec packets
copy    copy the DF bit from the inner header
IPSec config>template 1 df-bit clear
```

Use the **no** form of this command to restore the COPY default value.

“TEMPLATE [ID] MTU-THRESHOLD [INTEGER 256..2000]”

Specifies the lowest Maximum Transfer Unit (MTU) that must be reported to hosts protected by the router following ICMP/PMTU message processing. Default is 576 bytes. In most applications, this value is a compromise between the behavior of the network performing the fragmentation and a very small MTU. It also depends on the type of traffic the network is carrying. For further information, please see Section 6 of RFC 1191.

Example:

```
IPSec config>template 1 mtu-threshold 580
```

Use the **no** form of this command to restore the 576-bit default value.

“TEMPLATE [ID] MTU-DEFAULT {[INTEGER 256..2000]/DISABLED}”

Specifies the initial IPsec tunnel MTU that must be reported to the hosts protected by the router. This value is disabled by default and should only be assigned a value when the path MTU is known beforehand. For further information, please see Section 6 of RFC 1191.

Example:

```
IPSec config>template 1 mtu-default ?
<256..2000>    set starting value for path MTU
disabled       disables starting value for path MTU
```

```
IPSec config>template 1 mtu-default 512
```

Use the **no** form of this command to disable this setting.

Once all the corresponding parameters and keys have been defined, they would need be entered on the other end-point router with which we are going to establish the tunnel. And the last step would be to associate (map) the ACL entries to the templates, or in other words, create the SPD entries. We will come to this once we have explained how to configure the dynamic templates.

You can use the *LIST* and *NO* commands to view or delete configured templates:

Command	Operation
LIST TEMPLATE	Displays a list of all the elements on the template.
NO TEMPLATE	Deletes elements from the template.

“LIST TEMPLATE ALL”

Displays a list of all the elements on the template.

Example:

```
IPSec config>list template all
TEMPLATES
1 manual  ESP-DES  ESP-MD5  SRC=192.100.1.2 DES=192.100.1.1 SPI=280
2 manual  AH-SHA1  SRC=192.100.1.2 DES=192.100.1.10  SPI=280
```

“LIST TEMPLATE ADDRESS-FILTER [IP ADD] [MASK]”

Displays a list of all the elements on the template whose source/destination IP address fall within the range defined by the [IP ADD] and [MASK] fields.

Example:

```
IPSec config>list template address-filter 192.100.1.10  255.255.255.255
TEMPLATES
2 manual  AH-SHA1  SRC=192.100.1.2 DES=192.100.1.10  SPI=280
```

“NO TEMPLATE [ID]”

Deletes the template with the [ID] element as identifier.

Example:

```
IPSec config>no template 2
IPSec config>list template all
TEMPLATES
1 manual  ESP-DES  ESP-MD5  SRC=192.100.1.2 DES=192.100.1.1 SPI=280
```

2.3.2.2.2 Dynamic templates (IPsec IKE)

IPsec IKE (dynamic IPsec) requires two types of templates: dynamic templates, which are the same as the templates configured in manual mode, and, depending on the choice of negotiation protocol, either ISAKMP or IKEv2 templates. Now the tunnel endpoints need to negotiate the algorithms and keys that will be used to establish an IPsec Security Association. This is accomplished in two phases:

- In the first phase, the two tunnel endpoints authenticate one another and agree upon certain security parameters required to protect the negotiation. These parameters are defined in the ISAKMP templates or IKEv2 templates.
- Phase two is the negotiation of the security association for the tunnel, and is based on the dynamic templates.

In the following section, we describe the configuration commands available on ISAKMP templates, IKEv2 templates and then dynamic templates:

2.3.2.2.3 ISAKMP template parameters

We will begin by describing the ISAKMP parameters since they are the first step in the negotiations.

ISAKMP templates are configured using the following subcommands within the *TEMPLATE* command:

Command	Operation
ISAKMP	Creates an ISAKMP template with certain security parameters.
DESTINATION-ADDRESS	Includes the tunnel destination endpoint address in the template.
DISCOVER	Indicates that TED should be used to discover the tunnel endpoint address.
BACKUP-DESTINATION	Adds a backup destination address.

DISABLE-FAST-RE-TURN-FROM-BACKUP	Disables checking if Main is available and fast return to Main from Backup.
UDP-ENCAPSULATION	Encapsulates IPsec packets in UDP packets.
UDP-IKE	Encapsulates IPsec IKE packets in UDP packets.
LIFE	Adds the SAs lifetime created from the template.
IKE	Configures parameters relating to IPsec IKE mode.
KEEPALIVE	Enables/disables the available keepalive services.
CONFIG	Allows you to define whether the device will initiate the configuration method, wait for a proposal, or behave in accordance with the IKE method used. This command is described in *ISAKMP configuration mode* on page 63.
AGGRESSIVE	Configures aggressive mode as the encryption mode (encrypted/clear) for the third IKE message.
SEND-ORIGINAL-PKT	Once the tunnel is fully established, sends the original packet that led to the tunnel being created.
SET-LABEL	Packets processed by IPsec are tagged with this label.
FRONT-DOOR-VRF	Sets the VRF used to reach the tunnel destination.
NO	Negates a command or restores its default value.

The first thing to do is to configure the ISAKMP template security parameters that will be used to negotiate the SA connection. For its part, the ISAKMP template also gives rise to a SA negotiation, or ISAKMP SA:

“TEMPLATE [ID] ISAKMP [ENCRYPT] [AUTHEN]”

Creates an ISAKMP template based on encryption and authentication algorithms. The encryption options supported are DES, Triple DES (TDES), AES128, AES192, and AES256. The authentication options are MD5, SHA, SHA256, SHA384, and SHA512. The three AES encryption types use different key lengths (128, 192 and 256 bits, respectively). Despite being similar to ESP, this is not ESP and an authentication algorithm must be selected.

Example:

```
IPSec config>template 2 isakmp tdes sha1
```

Now we need to specify the tunnel endpoint address. ISAKMP templates do not require a source address.

“TEMPLATE [ID] DESTINATION [IP ADD/DOMAIN NAME]”

Example:

```
IPSec config>template 2 destination-address 192.100.1.1
```

In this particular case, 192.100.1.1 is used as the IP destination address.

Example:

```
IPSec config>template 4 destination-address XXX.bintec.de
```

In this case, the domain name XXX.bintec.de is used as the destination. Keep in mind that you will need to configure a DNS server to resolve the domain name.

We also have the option of specifying any address (0.0.0.0.) for the remote address and using the TED protocol to dynamically discover the remote tunnel endpoint address or waiting for the remote end to open the tunnel.

“TEMPLATE [ID] DISCOVER”

With this option enabled on the template and choosing 0.0.0.0 as the destination address, we would use a TED negotiation to establish the remote end of the tunnel. Before we configure the router to discover remote addresses, we should consider some of the limitations of the TED protocol (see chapter 1). This is disabled by default.

Example:

```
IPSec config>template 2 discover
```

Use the no form of this command to restore the default behavior and disable TED. This is useful when we are going to specify the remote tunnel endpoint address or wait for the remote tunnel endpoint to open the tunnel, or when we are using the *PKT-DEST-ISAKMP-DEST* advanced option (see below).

“TEMPLATE [ID] BACKUP-DESTINATION [IP ADD]”

Adds a backup destination IP address.

ISAKMP templates admit up to three backup addresses so that the router can establish a tunnel with one of the backup addresses if it can't to do so with the main address.

While connected to the backup address, the router continually polls the main address to see if it can establish a tunnel with it. If it can, it does so and also closes the session established with the backup address.

The address polling period is calculated as follows:

Main Address Polling Period: ADVANCED DPD IDLE-PERIOD + ADVANCED DPD PACKETS * ADVANCED DPD INTERVAL (seconds).

Which, with the default values, results in:

Main Address Polling Period: 60 + 5 * 3 = 75 seconds.

Example:

```
IPSec config>template 2 backup-destination 192.100.1.2
```

Use the **no** form of this command to delete the backup destination IP address.

“TEMPLATE [ID] DISABLE-FAST-RETURN-FROM-BACKUP”

Configuring this option changes the default behavior of the router when it is connected to a backup address. The router no longer polls the main address to see if it can establish a tunnel with it, so the session with the backup address remains open.

Example:

```
IPSec config>template 2 disable-fast-return-from-backup
```

Use the **no** form of this command to restore the default behavior (the router continually polls the main address to see if it can establish a tunnel with it).

Command history:

Release	Modification
11.01.08	The DISABLE-FAST-RETURN-FROM-BACKUP option was introduced in isakmp templates as of version 11.01.08.

And finally, there are a number of optional parameters whose default values can be modified if necessary:

“TEMPLATE [ID] UDP-ENCAPSULATION”

Specifies that IPsec packets are to be encapsulated in UDP. This is often used to cross firewalls or devices running NAPT without changing the configuration. By default, this option is not enabled.

Example:

```
IPSec config>template 2 udp-encapsulation
```

Use the **no** form of this command to restore the default behavior (IPsec packets are not encapsulated in UDP packets).

“TEMPLATE [ID] UDP-IKE”

Specifies that IPsec packets (IKE) are to be encapsulated in UDP. This is often used to cross firewalls or devices running NAPT without changing the configuration. By default, this option is not enabled.

Example:

```
IPSec config>template 2 udp-ike
```

Use the **no** form of this command to restore the default behavior (IPsec packets are not encapsulated in UDP packets even though data packets are).

“TEMPLATE [ID] LIFE DURATION SECONDS [VALUE]”

Allows you to introduce the negotiation SA lifetime. This is 3,600 seconds (1 hour) by default.

Example:

```
IPSec config>template 2 life duration seconds 1000
```

Use the **no** form of this command to delete the lifetime.

“TEMPLATE [ID] IKE”

Various parameters relating to IPsec IKE mode can be configured on ISAKMP templates:

Example:

```
IPSec config>template 2 ike ?
  ca                      CA
  mode                    Mode in which phase I of the ISAKMP/IKE exchange is
                          carried out
  method                  Establishes the authentication method used by the
                          device
  id                      Identifier used during phase 1 of the ISAKMP/IKE
                          exchange
  idtype                  Types of identifiers used during phase 1 of the
                          ISAKMP/IKE exchange
  crl                     CRL
  group                   group
  fragmentation           IKE Fragmentation
  lifetime-negotiation    Enables lifetime negotiation
  early-retry             Retry IKE negotiation in 1/4 of purgetime
  delayed-retry           Delay retries of non answered IKE negotiations
  delayed-retry-max       Maximum Delay of retries of non answered IKE
                          negotiations
  natt-version            Send natt vendor id specific version
  no                      Negates a command or set it default
IPSec config>
```

“TEMPLATE [ID] IKE MODE {AGGRESSIVE/MAIN}”

Phase I ISAKMP/IKE exchanges can be done in two modes: *main* mode or *aggressive* mode. Aggressive mode is faster than main mode, but fewer parameters can be negotiated. Main mode is enabled by default.

Example:

```
IPSec config>template 2 ike mode aggressive
```

Use the **no** form of this command to restore the default value.

“TEMPLATE [ID] IKE METHOD {PRESHARED/RSA/XAUTH-INIT-PRESHARED/XAUTH-INIT-RSA}”

Establishes the authentication method used by the device. The following authentication methods can be configured: pre-shared key, RSA, XAUTH init pre-shared key and XAUTH init RSA key. The default is pre-shared key authentication.

Example:

```
IPSec config>template 2 ike method ?
   preshared               Preshared key method
   rsa                     RSA method
   xauth-init-preshared    Xauth init preshared key method
   xauth-init-rsa          Xauth init rsa key method
IPSec config>TEMPLATE 2 ike method rsa
```

XAUTH-init-preshared and **XAUTH-init-RSA** authentication methods are explained in detail in *ISAKMP configuration mode* on page 63.

The *CA* and *CRL* IKE options are only available when RSA or XAUTH-INIT-RSA are selected. These authentication methods are explained in detail in *Obtaining certificates through CSR* on page 132 and in *Certificate revocation list (CRL)* on page 147, respectively.

Use the **no** form of this command to restore the default value.

“TEMPLATE [ID] IKE IDTYPE {IP/FQDN/UFQDN/KEYID/ASN-DN}”

You can select different types of identifiers for phase I of the ISAKMP /IKE exchange:

(See the *IKE ID* command)

IP: indicates that the device's own IP address will be used as the device identifier.

Fully Qualified Domain Name (FQDN): uses a string equivalent to a network interface TCP/IP address for identification. For example, if you set the host name to XXX1 and the domain name to bintec.de, the device's FQDN used for

identification purposes in IPsec will be XXX1.bintec.de. You can only use this option in AGGRESSIVE mode.

Note

If you fail to configure the domain name correctly, the device will use only the host name followed by a period (.), which stands for the root domain. This must be taken into account at the remote end.

User Fully Qualified Domain Name (UFQDN): includes a specific user within the device in the form of an SMTP mail address (*user@host.domain.xx*). In this case, since there aren't any users in the device, the router sends the same string that was sent in the previous case. You can only use this option in AGGRESSIVE mode.

KEYID: Identification is verified by means of a binary string used to send specific information about the router's manufacturer. We use the host name configured in the device for this The domain names and subdomain names are not considered. You can only use this method in AGGRESSIVE mode.

ASN-DN: specifies the main certificate's (of those certificates being exchanged in order to establish the SA) distinguished name (DN) in binary DER-encoded format, as defined in the ASN.1 X.500 standard.

The remote device will use the received identifier (ID) and search for it in its table of pre-shared keys associated with devices (IP addresses or host names), created with the *KEY IP/HOSTNAME* command (which we will come to later).

Example:

```
IPSec config>template 2 ike idtype keyid
```

Use the **no** form of this command to restore the default value.

“TEMPLATE [ID] IKE ID [NAME/IP ADD]”

Sets the identifier used in phase I of the ISAKMP/IKE exchange.

If the IKE IDTYPE IP parameter is configured in the template, it specifies an IP address. If not, it is a character string.

If you don't set this parameter, the identifier used in IKE phase I is the one specified above in the *IKE IDTYPE* command description.

Example:

```
Office IPSec config>template 2 ike idtype keyid
```

Here, the Office ID is used.

Example:

```
Office IPSec config>template 2 ike idtype keyid
Office IPSec config>template 2 ike id MyOffice
```

Here, MyOffice ID is used.

Example:

If the device's IP address is 10.0.0.1

```
Office IPSec config>template 2 ike idtype ip
```

The ID used in this case is IP 10.0.0.1.

Example:

If the device's IP address is 10.0.0.1

```
Office IPSec config>template 2 ike idtype ip
Office IPSec config>template 2 ike id 1.1.1.1
```

The ID used in this case is IP 1.1.1.1.

Use the **no** form of this command to delete the ID.

“TEMPLATE [ID] IKE GROUP {ONE/TWO/FIVE/FIFTEEN}”

Specifies the Oakley group type. The higher the group level, the longer the negotiation. This is because higher-level groups require more processing. Group 1 is the default value.

Example:

```
IPSec config>template 2 ike group one
```

Use the **no** form of this command to restore the default value.

“TEMPLATE [ID] IKE FRAGMENTATION {DISABLE/FORCE}”

Disables/forces fragmentation of IKE packets before they are sent. The behavior is as follows:

The device fragments IKE packets 5 and 6 of the IKE negotiation with RSA and XAUTH-INIT-RSA provided that:

- The remote end sends the corresponding payload vendor to show that it supports the feature and you have not set the DISABLE option for this command.
- You have set the FORCE option for this command.

Example:

```
IPSec config>template 2 ike fragmentation force
```

Use the **no** form of this command to restore the default value.

The device fragments IKE negotiation packets 5 and 6 with RSA and XAUTH-INIT-RSA provided that the remote end sends the corresponding payload vendor to show that it supports the feature.

Example:

```
IPSec config>template 2 ike no fragmentation
```

“TEMPLATE [ID] IKE LIFETIME-NEGOTIATION”

Allows the router to send a lifetime proposal in the negotiation, i.e., the device proposes a lifetime that is negotiated. This option is enabled by default.

If *IKE NO LIFETIME-NEGOTIATION* is configured, the device will not send a lifetime proposal. This is useful when:

- The router encounters a device that refuses to negotiate if there is a lifetime proposal.
- You want the remote end to set the lifetime.

Warning

Some devices will refuse to negotiate if the lifetime proposal parameter is disabled.

Example:

```
IPSec config>template 2 ike lifetime-negotiation
```

“TEMPLATE [ID] IKE EARLY-RETRY”

If there has been no reply, IKE negotiation is retried after a quarter of the PURGE-TIMEOUT interval has gone by. Configure this interval using the *ADVANCED PURGE-TIMEOUT* command.

This option is useful in WWAN environments where the first packets sent are often lost and where highly disperse traffic means packets never progress.

Example:

```
IPSec config>template 2 ike early-retry
```

“TEMPLATE [ID] IKE DELAYED-RETRY”

If there has been no reply, IKE negotiation is retried after 2^(number of retry) seconds plus the PURGE-TIMEOUT interval has gone by. By default this option is enabled but it has no efect in case "early-retry" is configured.

The interval between retries will be increasing till it reachs the maximun configured with *DELAYED-RETRY-MAX* command.

Example:

```
IPSec config>template 2 ike delayed-retry
```

Command history:

Release	Modification
11.00.10	The DELAYED-RETRY option was introduced on ike templates as of version 11.01.10.

“TEMPLATE [ID] IKE DELAYED-RETRY-MAX”

It is the maximun number of seconds between retries if IKE negotiation has no reply. By default it is 600 seconds (5 minutes).

This option is useful in WWAN environments where the first packets sent are often lost and where highly disperse traffic means packets never progress.

Example:

```
IPSec config>template 2 ike delayed-retry max 2m
```

Command history:

Release	Modification
11.01.10	The DELAYED-RETRY-MAX option was introduced on ike templates as of version 11.01.10.

“TEMPLATE [ID] IKE NATT-VERSION”

Lets the remote end of the NAT-Traversal (NATT) type know that the router wants to negotiate. When the remote end initiates negotiation, the router will adapt to whatever proposal the remote end makes as long as it supports the version that the remote end proposes. The supported versions are:

- RFC: rfc 3947
- DRAFT-V3: NAT-Traversal version draft 3.
- DRAFT-V2-N: NAT-Traversal version draft 2-n.
- DRAFT-V2: NAT-Traversal version draft 2.
- NONE: Disables the NAT-Traversal feature (the device will not tell the remote end that it supports NAT-Traversal).

Example:

```
IPSec config>template 2 ike natt-version draft-v3
```

Command history:

Release	Modification
11.00.05	As of version 11.00.05, the default value for NATT is RFC.
11.01.00	As of version 11.01.00, the default value for NATT is RFC.

“TEMPLATE[ID] KEEPALIVE DPD”

Enables Dead Peer Detection (DPD) for SA maintenance.

Example:

```
IPSec config>template 2 keepalive dpd
```

Use the **no** form of this command to disable DPD.

Command history:

Release	Modification
11.00.05	As of version 11.00.05, keepalive DPD is enabled by default.
11.01.00	As of version 11.01.00, keepalive DPD is enabled by default.

“TEMPLATE [ID] AGGRESSIVE {CIPHER/CLEAR}”

Specifies whether or not to encrypt the third message of an IKE aggressive mode negotiation. The default value is clear.

Example:

```
IPSec config>template 2 aggressive cipher
```

Use the **no** form of this command to restore the default value.

“TEMPLATE [ID] SEND-ORIGINAL-PKT”

An IPsec tunnel is created when a packet requiring encryption arrives or is generated and the corresponding tunnel doesn't exist yet. Said packet, the original packet, doesn't usually get sent through the tunnel once it is established. If

you configure this command, the original packet is saved and sent through the tunnel once the tunnel is set up. This command is not configured by default.

The packets received straight after the original packet (up to a maximum of eight) are also stored for later delivery.

Example:

```
IPSec config>template 2 send-original-pkt
```

This command is disabled by default and the original packet is not sent through the tunnel once it is established. When the *SEND-ORIGINAL-PKT* command is not configured, the original packet is only sent through the tunnel if that is normal procedure.

“TEMPLATE [ID] SET-LABEL ENCODED”

IPsec encoded packets are marked with a corresponding label configured with this option.

Only outbound IKE packets are labeled: inbound IKE packets are not labeled.

If the SET-LABEL ENCODED option is not configured in the ISAKMP template but is in the matching DYNAMIC template, outbound IKE packets are marked with the label configured in the DYNAMIC template.

Example:

```
IPSec config>template 2 set-label encoded 2
```

Use the **no** form of this command to disable packet marking.

“TEMPLATE [ID] FRONT-DOOR-VRF [VRF-NAME]”

Allows you to configure the tunnel destination's VRF domain. IKE negotiation packets and encrypted packets will then be routed using this VRF. The use of this command means that the VRF domains for traffic before and after IPsec encryption may be different.

Example:

```
IPSec config>template 2 front-door-vrf ?
  <word>         Text
  global         Set the global VRF
IPSec config>template 2 front-door-vrf vrf-ext-2
```

Use the **no** form of this command to delete the front door VRF associated with this dynamic template.

Command history:

Release	Modification
11.01.05	This command is available in ISAKMP and IKEv2 templates as of version 11.01.05.

All parameters related to ISAKMP are configured through this. When the router wants to set up a secure tunnel, the first thing it must do is send the appropriate ISAKMP template proposals to the other end (according to the destination IP address) so that both devices can agree on which template to use.

Once the SA negotiation process is complete, the dynamic template must be taken into account in the agreement in order to create the SA connection.

2.3.2.2.4 IKEv2 template parameters

IKEv2 templates are configured using the following subcommands within the *TEMPLATE* command:

Command	Operation
IKEV2	Creates an Ike version 2 template.
DESTINATION-ADDRESS	Inserts the remote tunnel endpoint address in the template.
DISCOVER	Uses TED to discover the remote end of the tunnel.
BACKUP-DESTINATION	Inserts the backup destination IP address in the template.
DISABLE-FAST-RE-TURN-FROM-BACKUP	Disables checking if Main is available and fast return to Main from Backup.
UDP-ENCAPSULATION	Enables UDP encapsulation.
UDP-IKE	Enables IKE UDP encapsulation.
LIFE	Inserts the SA lifetime created from the template.
IKE	Configures parameters relating to IPsec IKE mode.

KEEPALIVE	Enables/disables the available keepalive services.
CONFIG	Defines whether the device will initiate the configuration method. This command is described in *IKEv2 configuration mode* on page 74.
SEND-ORIGINAL-PKT	Sends the original packet once the tunnel is established.
SET-LABEL	This value is inserted in IKE and IPsec packets.
FRONT-DOOR-VRF	Sets the VRF used to reach the tunnel destination.
NO	Negates a command or sets its default.

In this section, we will describe the commands that mark the difference between ISAKMP and IKEv2 templates.

The first thing to do when configuring an IKEv2 template is to set the security parameters that will be used to negotiate the IPsec connection.

“TEMPLATE [ID] IKEV2 [ENCRYPTION]”

Specifies which type of encryption algorithm(s) the IKEv2 template should use. The options are: DES, Triple DES (TDES), AES128, AES192, AES256, AES128GCM128 and AES256GCM128. You can specify multiple options. Encryption algorithms that include authentication (e.g., any GCM variant) can not be configured in the same template with encryption-only algorithms.

Example:

```
IPSec config>template 2 ikev2 encryption tdes aes192
```

“TEMPLATE [ID] IKEV2 [AUTHENTICATION]”

Specifies which type of authentication algorithm(s) the IKEv2 template should use. The options are: MD5, SHA, SHA256, SHA384 and SHA512. You can specify multiple options. Templates with encryption algorithms that include authentication (e.g., any GCM variant) do not require the configuration of this command.

Example:

```
IPSec config>template 2 ikev2 authentication md5
```

“TEMPLATE [ID] IKEV2 [PRF]”

Specifies which type of pseudo-random function (PRF) algorithm the IKEv2 template should use. The options are MD5, SHA, SHA256, SHA384 and SHA512. You can specify multiple options.

If you fail to configure this, the values configured with the *AUTHENTICATION* command are used.

“TEMPLATE [ID] IKEV2 [GROUP]”

Specifies the Diffie-Hellman group algorithm. You can specify more than one between groups one, two, five and fifteen.

IKEv2 also allows the default mode, which doesn't configure encryption or authentication.

Example:

```
IPSec config>template 2 ikev2
```

The *LIST IKEV2 DEFAULT-PROPOSAL* command displays the default values of the IKEv2 proposal:

Example:

```
IPSec config>list ikev2 default-proposal
IKEv2 Default Proposal:
   Encryption: AES256 AES192 AES128 3DES DES
   Authentication: SHA512 SHA384 SHA256 SHA1 MD5
   PRF: SHA512 SHA384 SHA256 SHA1 MD5
   DH Group: 5 2

IPSec config>
```

“TEMPLATE [ID] IKE”

Configures certain IPsec IKE parameters in IKEv2 templates:

Example:

```
IPSec config>template 2 ike ?
  ca                      CA
  method                  Establishes the authentication method used by the
```

```
                          device
  id                      Identifier used during phase 1 of the ISAKMP/IKE
                          exchange
  idtype                  Types of identifiers used during phase 1 of the
                          ISAKMP/IKE exchange
  crl                     CRL
  fragmentation           IKE Fragmentation
  lifetime-negotiation    Enables lifetime negotiation
  early-retry             Retry IKE negotiation in 1/4 of purgetime
  natt-version            Send natt vendor id specific version
  window-size             Window size announced for overlapping requests
  no                      Negates a command or sets its default
IPSec config>
```

The following parameters differ from the ISAKMP parameters described above:

“TEMPLATE [ID] IKE METHOD {LOCAL/REMOTE}{PRESHARED/RSA}”

Specifies a local or remote authentication method for the device. The options are: pre-shared key and RSA. The default is pre-shared key authentication.

Example:

```
IPSec config>template 2 ike method remote rsa
```

The *CA* and *CRL* IKE options are only available when **rsa** is selected on the local or remote end.

Use the **no** form of this command to restore the default value.

"TEMPLATE [ID] IKE NATT-VERSION"

Informs the remote end of the NAT-Traversal (NATT) type that the router wants to negotiate. When the remote end initiates the negotiation, the router will adapt to whatever proposal the remote end makes as long as it supports the version that the remote end proposes. The supported versions are:

- RFC: rfc 3947
- NONE: Disables the NAT-Traversal feature (the device will not tell the remote end that it supports NAT-Traversal).

"TEMPLATE [ID] IKE WINDOW-SIZE"

Allows overlapping IKEv2 request-response messages, permitting the recipient to send multiple requests before getting a response to the first one that was sent.

Configured IKEv2 templates may be viewed or deleted using the *LIST* and *NO* commands:

Command	Operation
LIST TEMPLATE	Displays a list of templates and their elements.
NO TEMPLATE	Removes elements from the list of templates.

“LIST TEMPLATE ALL”

Displays a list of templates and their elements.

Example:

```
IPSec config>list template all
TEMPLATES
1 ikev2  SRC=internal DES=0.0.0.0
   Encryption: AES256
   Authentication: SHA256
   PRF: SHA256
   DH Group: (default) 5 2
   LifeTime:1h0m0s
   Local PRESHARED
   Remote PRESHARED
   addr4 ID TYPE
   DPD enabled

2 isakmp 3DES MD5  SRC=internal DES=0.0.0.0
   LifeTime:1h0m0s
   IKE MAIN
   PRESHARED
```

```
addr4 ID TYPE
OAKLEY GROUP 1
DPD enabl
```

Command history:

Release	Modification
11.00.07	The IKEv2 template, with its IKE options, was introduced as of version 11.00.07.
11.01.02	The IKEv2 template, with its IKE options, was introduced as of version 11.01.02.
11.01.08	Configuration of encryption algorithms AES256GCM128 and AES128GCM128 was included for the IKEv2 templates.

2.3.2.2.5 Dynamic template parameters

Dynamic templates are configured using the following subcommands in the *TEMPLATE* command:

Command	Operation
DYNAMIC	Creates a dynamic template with a security service (ESP or AH).
NEGOTIATION-PROTOCOL	Configures the phase I negotiation protocol (IKEv1/IKEv2) to be used.
SOURCE-ADDRESS	Inserts the tunnel endpoint source address in the template.
DESTINATION-ADDRESS	Inserts the tunnel endpoint destination address in the template.
ANTIREPLAY	Enables anti-replay in the template.
PADDING-CHECK	Checks whether the padding field in the IPsec header takes the value specified in the RFC.
LIFE	Inserts the SA lifetime created from the template.
IKE	Configures parameters relating to IPsec IKE.
KEEPALIVE	Enables/disables the available keepalive services.
ENCAP	Configures either tunnel or transport operation mode.
NAPT-ID-SKIPPED	IPsec must not mark packets for napt.
FAST-FORWARDER	Forces fast-forwarding of IPsec packets.
INVALID-SPI-RECOVERY	Enables invalid SPI reception notification regardless of whether there is an ISAKMP SA with the remote end.
DF-BIT	Specifies DF bit processing in IPsec packets.
MTU-THRESHOLD	Specifies the minimum MTU threshold for PMTU processing.
MTU-DEFAULT	Specifies the initial MTU value that is sent through the IPsec tunnel.
TCP-MSS-ADJUST	Adjusts the MSS field value of TCP packets in transit.
RRI-ENABLED	Enables Reverse Route Injection (RRI).
RRI-NEXTHOP	Configures the next hop that RRI must use.
RRI-METRIC	Metric used to install route by RRI.
MAPPED-TO-IFC	Maps a template to an interface.
FRONT-DOOR-VRF	Sets the VRF used to reach the tunnel destination.
DIRECT-DECODED-FWD	Sends decoded packets directly to a next hop or an interface (when forwarding).
DIRECT-ENCODED-FWD	Sends encoded packets directly to a next hop or an interface (when forwarding).
ASSIGNED-ADDRESS-GOES-TO-IFC	The address received during the ISAKMP configuration is established on this interface.
UNIQUE	Only one similar tunnel per entry on the access list.
PKT-SRC-CLIENT-SRC	Uses the original packet's original IP address as original client.
REPLACE-DESTINATION	Replaces the destination of encapsulated packets with the tunnel destination.
PREFRAGMENTATION	Enables data pre-fragmentation.
GDOI GROUP	Configures the dynamic template as a client in a GDOI group.
SET-LABEL	This label is inserted in packets processed by IPsec.
SET-SESSION-MARK	Set an AFS session-mark value to IKE and IPsec packets.
VRF	Assigns the template to a VRF.
FAULT-TOLERANT	Tunnels opened with this template participate in fault tolerant IPsec recovery.
REPORT	Reports IPsec statistics to a generic-input NSLA filter.
TUNNEL-PROTECTION	TNIP interfaces protected by IPsec template.

NO	Negates a command or sets its default.

“TEMPLATE [ID] DYNAMIC ESP [ENCRYPT] [AUTHEN]”

Specifies which type of ESP the dynamic template should use. The options are DES, TDES, AES128, AES192, AES256, AES256GCM128 and AES128GCM128 for encryption and SHA1, SHA256, SHA384, SHA512 and NONE for authentication. The difference between the three AES encryption types is their key length (128, 192 and 256 bits, respectively). Encryption algorithms that include authentication (e.g., any GCM variant) do not require configuration of the authentication algorithm.

Example:

```
IPSec config>template 3 dynamic esp tdes sha1
```

“TEMPLATE [ID] DYNAMIC AH [AUTHEN]”

Specifies which type of AH the dynamic template should use. The options are MD5, SHA1, SHA256, SHA384 and SHA512.

Example:

```
IPSec config>template 3 dynamic ah md5
```

“TEMPLATE [ID] DYNAMIC GDOI-KS-POLICY”

Specifies that a dynamic template will be used by a GDOI client (group member) with the policy sent by the key server. This option should only be used when the dynamic template is configured as a GDOI client using the *TEMPLATE [ID] GROUP* command (explained later in the manual). This is why the GDOI GROUP 1 option is automatically configured when you configure this option.

Example:

```
IPSec config>template 3 dynamic gdoi-ks-policy
IPSec config>show config
; Showing Menu and Submenus Configuration for access-level 15 ...
;
         template 3 dynamic gdoi-ks-policy
         template 3 gdoi group 1
;
IPSec config>
```

Command history:

Release	Modification
11.00.05	The GDOI-KS-POLICY option was introduced in dynamic templates as of version 11.00.05.
11.01.00	The GDOI-KS-POLICY option was introduced in dynamic templates as of version 11.01.00.
11.01.08	Configuration of encryption algorithms AES256GCM128 and AES128GCM128 was included for the dynamic templates.

“TEMPLATE [ID] NEGOTIATION-PROTOCOL [IKEV1|IKEV2]”

Configures the phase I negotiation protocol to use. The default is IKEv1.

Example:

```
IPSec config>template 3 negotiation-protocol ikev2
IPSec config>
```

Command history:

Release	Modification
11.00.07	The *NEGOTIATION-PROTOCOL* option was introduced as of version 11.00.07.
11.01.02	The *NEGOTIATION-PROTOCOL* option was introduced as of version 11.01.02.

“TEMPLATE [ID] SOURCE-ADDRESS [IP ADD/INTERFACE]”

Adds the tunnel's local IP address. You can also specify an interface from where the IP address is taken. You only need to define this for dynamic templates.

The address you add can be an unnumbered address. That is, you can add the address of an interface that is unknown at the time the device is configured and which will, for example, be assigned by another mechanism such as PPP.

If this is set to 0.0.0.0 (i.e., not configured), the source address used is the address of the outgoing interface.

Example:

```
IPSec config>template 3 source-address 192.100.1.2
```

“TEMPLATE [ID] DESTINATION-ADDRESS [IP ADD/DOMAIN NAME]”

Adds the remote tunnel endpoint address.

Example:

```
IPSec config>template 3 destination-address 192.100.1.1
```

In this particular case, the IP destination address is 192.100.1.1.

Example:

```
IPSec config>template 3 destination-address XXX.bintec.de
```

In this case, the XXX.bintec.de domain name is used as the destination. Keep in mind that you will need to configure a DNS server to resolve the domain name.

If the remote tunnel endpoint address is set to 0.0.0.0., it is considered an unknown address and is not taken into account when selecting the dynamic template during negotiation. Since the destination address is unknown, only the remote end can begin IKE negotiation.

The following subcommands refer to established default values, but you might need to change them depending on the situation.

“TEMPLATE [ID] ANTIREPLAY”:

Enables anti-replay. This is a security method to protect against replay attacks. The command is enabled by default but can be disabled using the **no** form of this command.

Example:

```
IPSec config>template 3 no antireplay
```

“TEMPLATE [ID] PADDING-CHECK”

Although the old IPsec RFC allowed the padding field of the IPsec header to be filled with a random value, the current one specifies a particular value for said field. In order for the router to be able to work with devices still complying with the old RFC, you can configure a parameter that tells the router to check whether the padding field should take the value defined in the RFC or whether to ignore those data.

Example:

```
IPSec config>template 3 padding-check
```

Use the **no** form of this command to restore the default value (the IPsec header padding field will not be checked).

“TEMPLATE [ID] LIFE TYPE {SECONDS/KBYTES/BOTH}”

Allows you to enter the security association (SA) lifetime type based on the dynamic template. The lifetime in dynamic templates can be expressed as a time limit (SECONDS), as was the case for the ISAKMP templates, or as a limit on the number of bytes (KBYTES) transmitted through the SA generated with this template.

The third option (*BOTH*) lets you establish both limits at the same time. In this case, the SA is deleted when one of the limits expires. The default value is seconds.

Example:

```
IPSec config>template 3 life type both
```

Use the **no** form of this command to restore the default value.

“TEMPLATE [ID] LIFE DURATION {SECONDS/KBYTES} [VALUE]”

The chosen lifetime is given in the VALUE field. If you choose BOTH in the previous subcommand, you have to type the subcommand twice to give both types of values (seconds and kilobytes).

Example:

```
IPSec config>template 3 life duration seconds 20000
IPSec config>template 3 life duration kbytes 1000
```

Use the **no** form of this command to delete the lifetime.

“TEMPLATE [ID] IKE”

Configures certain IPsec/IKE parameters in dynamic templates:

Example:

```
IPSec config>template 3 ike ?
  pfs                    Enables the Perfect Forward Secrecy service
  group                  group
  lifetime-negotiation   Enables lifetime negotiation
  pkt-dest-isakmp-dest   Packet destination gives end tunnel destination
  no                     Negates a command or set it default
IPSec config>
```

“TEMPLATE [ID] IKE PFS”

Enables Perfect Forward Secrecy (PFS). This service both improves the security of the SAs and provides improved key management.

Example:

```
IPSec config>template 3 ike pfs
```

Use the **no** form of this command to disable PFS.

“TEMPLATE [ID] IKE GROUP {ONE/TWO/FIVE/FIFTEEN}”

Specifies the Oakley group type. The higher the group level, the longer the negotiation. This is because higher-level groups require more processing. Group 1 is the default value.

Example:

```
IPSec config>template 3 ike group one
```

Use the **no** form of this command to restore the default value.

“TEMPLATE [ID] IKE LIFETIME-NEGOTIATION”

Allows the router to send a lifetime proposal in the negotiation (i.e., the router proposes a lifetime that is negotiated). This option is enabled by default.

The router does not send a lifetime proposal when *IKE NO LIFETIME-NEGOTIATION* is configured. This is useful when:

- The router encounters a device that won't negotiate if there is a lifetime proposal.
- You want the remote end to set the lifetime.

Warning

Some devices won't negotiate when the lifetime proposal parameter is disabled.

Example:

```
IPSec config>template 3 ike lifetime-negotiation
```

“TEMPLATE [ID] IKE PKT-DEST-ISAKMP-DEST”

The packet destination specifies the tunnel remote IP address. This way, there is no need for you to configure a template destination address.

Example:

```
IPSec config>template 3 ike pkt-dest-isakmp-dest
```

Use the **no** form of this command to restore the default behavior.

“TEMPLATE [ID] KEEPALIVE KEEPALIVE”

Enables the keep-alive service for SA maintenance.

Example:

```
IPSec config>template 3 keepalive keepalive
```

Use the **no** form of this command to disable the keep-alive function.

“TEMPLATE [ID] ENCAP {TUNNEL/TRANSPORT}”

Specifies whether encapsulation will be done in tunnel or transport mode. Default is tunnel mode.

Example:

```
IPSec config>template 3 encap transport
```

Use the **no** form of this command to restore the default encapsulation mode.

“TEMPLATE [ID] NAPT-ID-SKIPPED”

Applies packet marking to IPsec traffic for NAPT. Marking is enabled by default.

Example:

```
IPSec config>template 3 napt-id-skipped
```

Use the **no** form of this command to restore the default behavior and apply packet marking.

“TEMPLATE [ID] FAST-FORWARDER”

Routes packets via a fast path to speed up routing. You shouldn't use this option with dynamic templates if you are going to apply additional processing (such as NAT) to IPsec traffic either before or after encapsulation.

Please note that you can't use this command when the *DIRECT-DECODED-FWD* and *DIRECT-ENCODED-FWD* commands are configured.

Example:

```
IPSec config>template 3 fast-forwarder
```

Use the **no** form of this command to restore the default behavior and disable fast-forwarding of IPsec packets.

“TEMPLATE [ID] INVALID-SPI-RECOVERY”

Sends an invalid SPI packet notification, regardless of whether there is an ISAKMP SA with the remote end. If an invalid SPI is received when there is no ISAKMP SA with the remote end, a new ISAKMP SA is created alerting the remote end of the invalid SPI and allowing it to delete the SA with that SPI.

Regardless of the configuration of this parameter, if there is an ISAKMP SA with the remote end when the invalid SPI is received, the receiver sends a report.

This function is enabled by default. Use the **no** form of this command to disable it.

Example:

```
IPSec config>template 3 no invalid-spi-recovery
```

“TEMPLATE [ID] DF-BIT {SET/CLEAR/COPY}”

When encapsulating a packet in IPsec, it is the router's job to ensure that the Path Maximum Transmission Unit Discovery (PMTUD) algorithm continues to function in hosts protected by the tunnel. The PMTUD algorithm uses the *don't fragment* (DF) bit in the IP header and ICMP packets. This option lets you specify the way in which the router processes the DF bit in IPsec packets: *always mark* (all IPsec traffic will leave with the DF bit set to TRUE), *always remove* (the bit is set to FALSE, ICMP/PMTUD packets are not processed and therefore the router acts as a black hole for this algorithm), or *copy* the packet being protected (normal ICMP/PMTUD processing and the router's default option). For further information, please see section 6 of RFC 1191.

Example:

```
IPSec config>template 3 df-bit ?
set     set the DF bit on the IPSec packets
clear   clear the DF bit on the IPSec packets
copy    copy the DF bit from the inner header
IPSec config>template 3 df-bit clear
```

Use the **no** form of this command to restore the default value *COPY*.

“TEMPLATE [ID] MTU-THRESHOLD [INTEGER 256..2000]”

Specifies the lowest maximum transfer unit (MTU) that the router must report to the hosts it protects as a result of ICMP/PMTU message processing. The default value is 576 bytes, since in most applications this is a compromise value between the behavior of the network performing fragmentation and a very small MTU. In any case, it depends on the type of traffic the network is carrying. For further information, please see section 6 of RFC 1191.

Example:

```
IPSec config>template 3 mtu-threshold 580
```

Use the **no** form of this command to restore the default value of 576 bytes.

“TEMPLATE [ID] MTU-DEFAULT {[INTEGER 256..2000]/DISABLED}”

Specifies the initial IPsec tunnel MTU that the router must report to the hosts it protects. This setting is disabled by default and should only be assigned a value when the MTU path is known beforehand. For further information, please see section 6 of RFC 1191.

Example:

```
IPSec config>template 3 mtu-default ?
<256..2000>    set starting value for path MTU
disabled       disables starting value for path MTU
IPSec config>template 3 mtu-default 512
```

Use the **no** form of this command to disable it.

“TEMPLATE [ID] TCP-MSS-ADJUST {[INTEGER 536..65535]/CLAMPING [HEADER_LENGTH]}”

Specifies the maximum TCP packet segment size (MSS) for a security association (SA) generated with this template.

The MSS value is based on the SA MTU size. Use the CLAMPING option to specify a value to subtract from the SA MTU. The SA MTU value is the lowest of the following values:

- Value configured with the DEFAULT-MTU option.
- Value learned by PMTUD.
- MTU value of the interface mapped with the MAPPED-TO-IFC option.

HEADER_LENGTH is 40 bytes by default.

Example:

```
IPSec config>template 3 tcp-mss-adjust ?
  <536..65535>    Adjust the mss of transit packets
  clamping       Automatically adjust the mss
IPSec config>template 3 tcp-mss-adjust clamping ?
  <cr>             Typical TCP/IP header length (40 bytes)
  header-length    Specify TCP/IP header length
```

When an integer is specified, it indicates the value the TCP SYN packet's MSS option will change to, provided it is lower than the value already in the packet.

Example 1:

```
IPSec config>template 3 tcp-mss-adjust 1100
```

This option allows you to limit the TCP MSS value in SAs generated with this template to 1100 bytes.

Example 2:

```
IPSec config>template 3 tcp-mss-adjust clamping
```

Limits the TCP MSS value in SAs generated with this template to the MTU size minus 40 bytes (default value).

Example 3:

```
IPSec config>template 3 tcp-mss-adjust clamping header-length 60
```

Limits the TCP MSS value in SAs generated with this template to the MTU size minus 60 bytes.

Use the **no** form of this command to restore the default behavior and disable the TCP MSS option for TCP packets transiting an SA generated with this template.

“TEMPLATE [ID] RRI-ENABLED”

Enables Reverse Route Injection (RRI) in the dynamic template. This creates a static route in the routing table when

this template is used to open a tunnel. The destination of the static route is the network (or subnet, or host) specified by the negotiated remote clients and the next hop value depends on the option selected via the *RRI-NEXTHOP* command (see below). This route lasts while the tunnel is open and can be broadcast normally using traditional dynamic routing algorithms (RIP, OSPF, etc.).

Example:

```
IPSec config>template 3 rri-enabled
```

Use the **no** form of this command to disable RRI in the dynamic template.

“TEMPLATE [ID] RRI-NEXTHOP {SOURCE/DESTINATION/USER-DEFINED[IP ADD]}”

Configures the next hop destination on RRI static routes. There are three possibilities:

SOURCE (default value) - the local tunnel endpoint address is used as the next hop for the static route. This is used specifically when the tunnel source interface is point-to-point or when all the traffic using the route is going to be encrypted and the remote end is different from the negotiated client.

DESTINATION - the remote endpoint address is used. This is used specifically when the tunnel endpoints are directly connected or when recursive routing is used to resolve an address rather than the default route.

USER-DEFINED [IP ADD] - the user specifies the IP address of the next hop. This is used in specific cases that do not fit into either of the above scenarios.

Example:

```
IPSec config>template 3 rri-nexthop ?
  source          tunnel source address
  destination     tunnel destination address
  user-defined    user-defined next hop
IPSec config>template 3 rri-nexthop user-defined ?
  <a.b.c.d>    Ipv4 format
IPSec config>template 3 rri-nexthop user-defined 10.10.10.1
```

Use the **no** form of this command to restore default value SOURCE.

“TEMPLATE [ID] RRI-METRIC [INTEGER 1..255]”

Configures the cost for the static route installed by Reverse Route Injection (RRI). Default cost for routes installed by RRI is 10.

Example:

```
IPSec config>template 3 rri-metric 10
```

Use the **no** form of this command to restore the default value.

Command history:

Release	Modification
11.00.05	This command was introduced as of version 11.00.05.
11.01.00	This command was introduced as of version 11.01.00.

“TEMPLATE [ID] MAPPED-TO-IFC”

Associates the dynamic template with an interface. This way, the router understands that it must only apply the template if traffic is sent through said interface.

Example:

```
IPSec config>template 3 mapped-to-ifc ppp1
```

Using this command has many implications, so we recommend that you gain a thorough understanding of how it works before you configure it. To provide some clarification of the various concepts at issue here, we will now give a few examples and explain how they work.

Example 1:

```
   feature access-lists
; -- Access Lists user configuration --
      access-list 100
         entry 1 default
         entry 1 permit
```

```
        entry 1 source address 10.127.0.28 255.255.255.255
        entry 1 destination address 10.127.1.29 255.255.255.255
;
      exit
;
   exit
;
;
   protocol ip
; -- Internet protocol user configuration --
      ipsec
; -- IPSec user configuration --
        enable
        assign-access-list 100
;
        template 1 isakmp tdes md5
        template 1 destination-address 192.168.169.29
        template 1 ike natt-version rfc
        template 1 keepalive dpd
;
        template 2 dynamic esp tdes md5
        template 2 source-address serial0/0
        template 2 destination-address 192.168.169.29
;
        map-template 100 2
        advanced dpd no always-send
      exit
;
   exit
;
```

In *Example 1*, all traffic with 10.127.0.28 as source and 10.127.1.29 as destination is protected by IPsec (irrespective of the interface through which it is sent). That is to say:

- Packets exiting the device with 10.127.0.28 as source and 10.127.1.29 as destination are **encapsulated** in an IPsec tunnel.
- Packets entering the device with 10.127.1.29 as source and 10.127.0.28 as destination must be **encapsulated** in an IPsec tunnel or the packet will be dropped.

By mapping template 2 to the serial0/0 interface, we get *Example 2*.

Example 2:

```
   feature access-lists
; -- Access Lists user configuration --
      access-list 100
        entry 1 default
        entry 1 permit
        entry 1 source address 10.127.0.28 255.255.255.255
        entry 1 destination address 10.127.1.29 255.255.255.255
;
      exit
;
   exit
;
;
   protocol ip
; -- Internet protocol user configuration --
      ipsec
; -- IPSec user configuration --
        enable
        assign-access-list 100
;
        template 1 isakmp tdes md5
        template 1 destination-address 192.168.169.29
        template 1 ike natt-version rfc
        template 1 keepalive dpd
;
```

```
         template 2 dynamic esp tdes md5
         template 2 source-address serial0/0
         template 2 destination-address 192.168.169.29
         template 2 mapped-to-ifc serial0/0
;
         map-template 100 2
         advanced dpd no always-send
      exit
;
   exit
;
```

In *Example 2*, **only** the traffic **sent through the serial0/0 interface** with 10.127.0.28 as source and 10.127.1.29 as destination is protected by IPsec. That is to say:

- Packets exiting the device through the serial 0/0 interface with 10.127.0.28 as source and 10.127.1.29 as destination are **encapsulated** in an IPsec tunnel.
- Packets exiting the device through an interface other than the serial 0/0 interface with 10.127.0.28 as source and 10.127.1.29 as destination do so in **clear**.
- Packets entering the device through the serial0/0 interface with 10.127.1.29 as source and 10.127.0.28 as destination must be **encapsulated** in an IPsec tunnel or the packet will be dropped.
- Packets entering the device through an interface other than the serial0/0 interface with 10.127.1.29 as source and 10.127.0.28 as destination **are sent in the usual way**, even if they haven't been encapsulated in IPsec.

If we now add a template that is not mapped to any interface, we get *Example 3*.

Example 3:

```
   feature access-lists
; -- Access Lists user configuration --
      access-list 100
         entry 1 default
         entry 1 permit
         entry 1 source address 10.127.0.28 255.255.255.255
         entry 1 destination address 10.127.1.29 255.255.255.255
;
      exit
;
   exit
;
;
   protocol ip
; -- Internet protocol user configuration --
      ipsec
; -- IPSec user configuration --
         enable
         assign-access-list 100
;
         template 1 isakmp tdes md5
         template 1 destination-address 192.168.169.29
         template 1 ike natt-version rfc
         template 1 keepalive dpd
;
         template 2 dynamic esp tdes md5
         template 2 source-address serial0/0
         template 2 destination-address 192.168.169.29
         template 2 mapped-to-ifc serial0/0
;
         template 3 dynamic esp tdes md5
         template 3 source-address serial0/0
         template 3 destination-address 192.168.169.29
;
         map-template 100 2
         map-template 100 3
         advanced dpd no always-send
      exit
;
   exit
```

```
;
```

In *Example 3*, traffic sent through the serial0/0 interface with 10.127.0.28 as source and 10.127.1.29 as destination is protected with IPsec in tdes and md5 mode, **as is the other traffic**. That is to say:

- Packets exiting the device through the **serial0/0** interface with 10.127.0.28 as source and 10.127.1.29 as destination are **encapsulated** in an IPsec tunnel.
- Packets exiting the device through an interface **other than** the serial 0/0 interface with 10.127.0.28 as source and 10.127.1.29 as destination are **encapsulated** in an IPsec tunnel.
- Packets entering the device through the **serial0/0** interface with 10.127.1.29 as source and 10.127.0.28 as destination must be **encapsulated** in an IPsec tunnel or the packet will be dropped.
- Packets entering the device through an interface **other than** the serial0/0 interface with 10.127.1.29 as source and 10.127.0.28 as destination must be **encapsulated** in an IPsec tunnel or the packet will be dropped.

Finally, *Example 4* is a sample configuration that allows the same traffic to be sent through two different tunnels. The router decides which tunnel to use based on the IP routing information. You generally use this configuration when you have a device with two interfaces, where one backs up the other, and the other interfaces (such as LAN or the like) are already protected against insecure access.

Example 4:

```
   feature access-lists
; -- Access Lists user configuration --
      access-list 100
         entry 1 default
         entry 1 permit
         entry 1 source address 10.127.0.28 255.255.255.255
         entry 1 destination address 10.127.1.29 255.255.255.255
;
      exit
;
   exit
;
;
   protocol ip
; -- Internet protocol user configuration --
      ipsec
; -- IPSec user configuration --
         enable
         assign-access-list 100
;
         template 1 isakmp tdes md5
         template 1 destination-address 192.168.169.29
         template 1 ike natt-version rfc
         template 1 keepalive dpd
;
         template 2 dynamic esp tdes md5
         template 2 source-address serial0/0
         template 2 destination-address 192.168.169.29
         template 2 mapped-to-ifc serial0/0
;
         template 3 isakmp tdes md5
         template 3 destination-address 1.1.1.2
         template 3 ike natt-version rfc
         template 3 keepalive dpd
;
         template 4 dynamic esp tdes sha1
         template 4 source-address ppp1
         template 4 destination-address 1.1.1.2
         template 4 mapped-to-ifc ppp1
         map-template 100 2
         map-template 100 4
         advanced dpd no always-send
      exit
;
   exit
;
```

In *Example 4*, if the IP routes indicate that traffic to 10.127.1.29 should go through the serial0/0 interface, all traffic with 10.127.0.28 as source and 10.127.1.29 as destination is protected with IPsec in tdes and md5 mode through a tunnel with the serial interface IP address as source and 192.168.169.29 as destination. If the IP routes change state and indicate that traffic to 10.127.1.29 should go through the ppp1 interface, this traffic is protected with IPsec in tdes and sha1 mode through a tunnel with the ppp1 interface IP address as source and 1.1.1.2 as destination.

Clearly, both tunnels can remain up at the same time and traffic can be sent through one and return through the other if the IP routes indicate this.

Important

If the template is mapped to an interface and also has **fast-forwarding** enabled, the packet is sent directly to the interface when it is encapsulated by IPsec. That is, it doesn't go back in the fast forwarder queue for routing as it usually would. Therefore, it doesn't follow the same scheme as described in the section on *Packet Processing with IPsec* .

Not only does this configuration allow packets to move faster (since they are sent over a faster path), it also allows you to give them special treatment once they have been encapsulated by IPsec. For example, you can get the packets to exit without performing NAT, or you can give them the outbound route before encapsulation, etc.

Please see the *FAST-FORWARDER* command for further information.

Use the **no** form of this command to remove the dynamic template association with the interface.

“TEMPLATE [ID] FRONT-DOOR-VRF [VRF-NAME]”

Allows you to configure the tunnel destination's VRF domain. IKE negotiation packets and encrypted packets will then be routed using this VRF. The use of this command means that the VRF domains for traffic before and after IPsec encryption may be different.

Example:

```
IPSec config>template 3 front-door-vrf ?
  <word>         Text
  global         Set the global VRF
IPSec config>template 3 front-door-vrf vrf-ext-2
```

If no value is configured, the VRF set by the *TEMPLATE <id> VRF <vrf name>* command is used as the front door VRF.

Use the **no** form of this command to remove the front door VRF associated with this dynamic template.

Command history:

Release	Modification
11.01.03	This command was introduced.

“TEMPLATE [ID] DIRECT-DECODED-FWD

Forwards the decoded packets directly to the configured next hop or interface. This command is not compatible with the *FAST-FORWARDER* command and only one of them can be configured.

Example:

```
IPSec config>template 3 direct-decoded-fwd ?
  <a.b.c.d>      Ipv4 format
  <interface>    Interface name
IPSec config>template 3 direct-decoded-fwd 20.20.20.2
```

Note

If you configure the router to forward the decoded packets to an interface rather than an IP address, make sure it is a point-to-point interface. If the outgoing interface is not point-to-point, please configure the next hop IP address.

Use the **no** form of this command to restore the default behavior.

Important

If you configure this command, the packet is sent directly to the interface when it is decapsulated by IPsec. That is, it does not go back in the forwarder queue for routing as it usually would. Therefore, it doesn't follow the same scheme as described in the section on *Packet Processing with IPsec* .

Command history:

Release	Modification
11.00.05	This command was introduced as of version 11.00.05.
11.01.00	This command was introduced as of version 11.01.00.

“TEMPLATE [ID] DIRECT-ENCODED-FWD

Forwards the encoded packets directly to the next hop or interface configured. This command is not compatible with the *FAST-FORWARDER* command and only one of them can be configured.

Example:

```
IPSec config>template 3 direct-encoded-fwd ?
  <a.b.c.d>      Ipv4 format
  <interface>    Interface name
IPSec config>template 3 direct-encoded-fwd 10.10.10.2
```

Note

If you configure the router to forward the encoded packets to an interface rather than an IP address, make sure it is a point-to-point interface. If the outgoing interface isn't point-to-point, please configure the next-hop IP address.

Use the **no** form of this command to restore the default behavior.

Important

If you configure this command, the packet is sent directly to the interface when it is encapsulated by IPsec. That is, it doesn't go back in the forwarder queue for routing as it usually would. Therefore, it doesn't follow the same scheme as described in the section on *Packet Processing with IPsec* .

Command history:

Release	Modification
11.00.05	This command was introduced as of version 11.00.05.
11.01.00	This command was introduced as of version 11.01.00.

“TEMPLATE [ID] ASSIGNED-ADDRESS-GOES-TO-IFC”

Establishes the address received during the ISAKMP configuration as the main address in the interface configured in this option.

The behavior is similar to the one described under *ASSIGNED IP ADDRESS DESTINATION* , when explaining the *ADVANCED ADDRESS-ASSIGNED-TO-IFC* command. However, in this case, it only takes effect if this template was selected during negotiation.

That is, unlike the *ADVANCED ADDRESS-ASSIGNED-TO-IFC* command, with this option you can select which interface the address will be established in when you have more than one alternative.

Example:

```
IPSec config>template 3 assigned-address-goes-to-ifc loopback1
```

Use the **no** form of this command to restore the default behavior.

“TEMPLATE [ID] UNIQUE”

Prevents two or more similar tunnels from being associated with the same access control list (ACL) entry. This option is disabled by default.

Example:

```
IPSec config>template 3 unique
```

Since you can map an ACL to various templates, you must configure the same value for all templates using the same ACL.

During the IPsec tunnel creation process, a new dynamic DX entry (relating to one of the static E entries on the list) is installed in the access control list. The IPsec tunnel associates itself exclusively with the dynamic DX entry. If the unique option is set, the router starts searching among the other dynamic entries relating to that E entry once the tunnel is fully established. If it finds a previous dynamic DY entry similar to the DX entry, it removes the tunnel associated with the DY entry. The DY entry ends up being deleted afterwards. The router considers the following characteristics to determine whether the DX and DY entries are similar: same source and destination, protocol, ports, VRF and action.

The following example illustrates how it works:

Example 1:

```
   feature access-lists
; -- Access Lists user configuration --
      access-list 100
         entry 1 default
         entry 1 permit
         entry 1 source address 10.122.1.0 255.255.255.0
         entry 1 destination address 10.121.1.0 255.255.255.0
;
      exit
;
   exit
;
;
   protocol ip
; -- Internet protocol user configuration --
      ipsec
; -- IPSec user configuration --
         enable
         assign-access-list 100
;
         template 1 isakmp tdes md5
         template 1 destination-address 192.168.121.8
         template 1 ike natt-version rfc
         template 1 keepalive dpd
;
         template 2 dynamic esp tdes md5
         template 2 destination-address 192.168.121.8
         template 2 unique
;
         template 3 isakmp tdes md5
         template 3 destination-address 192.168.121.5
         template 3 ike natt-version rfc
         template 3 keepalive dpd
;
         template 4 dynamic esp tdes md5
         template 4 destination-address 192.168.121.5
         template 4 unique
;
         map-template 100 2
         map-template 100 4
         advanced dpd no always-send
      exit
;
   exit
;
```

According to the configuration in *Example 1*, the router could set up a tunnel to encapsulate traffic between networks 10.121.1.0/24 and 10.122.1.0/24 where the outgoing interface is either ethernet0/0.10 or ethernet0/0.20. You can't have both tunnels up at the same time. If a tunnel is created with one of these interfaces as its local interface, the tunnel using the other Ethernet interface as its local interface (if there is one) is eliminated. The newly created tunnel connection always prevails over the older one.

Use the **no** form of this command to remove the restriction that prevents similar tunnels from being associated with the same ACL entry.

“TEMPLATE [ID] PKT-SRC-CLIENT-SRC”

When a locally sourced tunnel is being set up, the router checks to see whether this command has been configured. If it has, the router uses the packet's source IP address that led to the creation of the tunnel as the local client for the new tunnel. This way, individual tunnels can be set up for multiple clients and included in larger ACLs (rather than having to specify an ACL for each one). Moreover, individual tunnels can be managed separately despite sharing an ACL. This command is disabled by default.

Example:

```
IPSec config>template 3 pkt-src-client-src
```

Use the **no** form of this command to restore the default behavior.

“TEMPLATE [ID] REPLACE-DESTINATION”

Configures the dynamic template to replace the destination of IPsec encrypted packets with the IPsec tunnel destination. You should only use this when the tunnel is configured in transport mode (i.e., when the IP header of the IPsec packet is not encapsulated).

Example:

```
IPSec config>template 3 replace-destination
```

Use the **no** form of this command to restore the default behavior and stop the template replacing the destination of encapsulated packets.

“TEMPLATE [ID] PREFRAGMENTATION”

Enables packet fragmentation prior to SA encapsulation. The lowest of the following values is used for the SA MTU value:

- Value configured with the DEFAULT-MTU option.
- Value learned by PMTUD.
- MTU value of the interface mapped with the MAPPED-TO-IFC option.

Example:

```
IPSec config>template 3 prefragmentation
```

When this option is selected, any packets larger than the SA MTU size that have the *don’t fragment* bit in the IP header turned on are dropped. A PMTUD packet is sent to the source regardless of the value configured using the *TEMPLATE [ID] DF-BIT* command.

Packet fragmentation is not compatible with encapsulation in TEMPLATE [ID] ENCAP TRANSPORT transport mode.

Use the **no** form of this command to restore the default behavior and stop packets from being fragmented prior to SA encapsulation.

“TEMPLATE [ID] GDOI GROUP [VALUE]”

Configures the dynamic template as a client in a GDOI group. The value configured is the GDOI group ID. The GDOI server address is the destination configured in the template.

Example:

```
IPSec config>template 3 gdoi group 1
```

Use the **no** form of this command to disable GDOI in the dynamic template.

“TEMPLATE [ID] SET-LABEL {ENCODED/DECODED} [VALUE]”

Configures the label used to mark IPsec encoded/decoded packets.

IKE packets are only marked at egress with the label configured using the SET-LABEL ENCODED option in the ISAKMP template. Received IKE packets are not marked.

If the SET-LABEL ENCODED option is not configured in the ISAKMP template but is in the matching DYNAMIC template, the IKE packets are marked at egress with the label configured in the DYNAMIC template.

Example:

```
IPSec config>template 3 set-label encoded 2
```

Use the **no** form of this command to disable packet marking.

“TEMPLATE [ID] SET-SESSION-MARK DECODED [INTEGER 0..65535]”

Configures the label used in AFS sessions for IPsec decoded packets.

Example:

```
IPSec config>template 3 set-session-mark decoded 100
```

Use the **no** form of this command to disable AFS session marking.

Command history:

Release	Modification
11.00.05	This command was introduced as of version 11.00.05.
11.01.00	This command was introduced as of version 11.01.00.

“TEMPLATE [ID] TUNNEL-PROTECTION <tnipX> [SHARED]”

Allows you to configure IP tunnel protection using an IPsec template.

Example:

```
IPSec config>template 2 tunnel-protection tnip1
```

Using the **SHARED** keyword allows you to protect several tnip configured to run in gre mode (ip or multipoint) with the same source and destination.

If **TUNNEL-PROTECTION** is configured on the template, then the template takes the source address, internal vrf (VRF) and external VRF (front-door-vrf) from the tnip interface.

Note

When you configure **TUNNEL-PROTECTION** on the template, the SOURCE-ADDRESS- MAPPED-TO-IFC, DIRECT-ENCODED-FWD, DIRECT DECODED-FWD, VRF and FRONT-DOOR-VRF options must not be configured and the template must not be mapped to any ACLs.

Command history:

Release	Modification
11.01.04	This command was introduced as of version 11.01.04.
11.01.05	As of version 11.01.05, IPsec templates with tunnel protection take VRFs from the tnip interface (meaning they cannot be configured).

“TEMPLATE [ID] VRF”

Assigns the dynamic template to a VRF. This means that the router only applies the template to traffic sent through the VRF.

Example:

```
IPSec config>template 3 vrf VRF1
```

If this command is configured and the template is associated with an interface (through the *TEMPLATE [ID] MAPPED-TO-IFC* command), said interface must also be associated with the VRF that you want to assign the template to. Otherwise, you'll get an error message. So, for example, if the ethernet0/0 interface is associated with VRF1, and the dynamic template is associated with the ethernet0/0 interface, an error appears if you try to assign the template to a VRF other than VRF1.

Example:

```
IPSec config>template 3 vrf VRF2
CLI Error: The mapped interface is not associated to this vrf
CLI Error: Command error
IPSec config>
```

Below we'll use some examples to help clarify these concepts and explain how the commands work:

The router in these examples has been configured with two VRFs, named VRF1 and VRF2. Two Ethernet subinterfaces have been created on the router, each associated with one of the VRFs, and both with the same IP address.

```
   network ethernet0/0.10
; -- Ethernet Subinterface Configuration --
      ip vrf forwarding VRFE1
;
      ip address 192.168.212.201 255.255.254.0
;
;
;
      encapsulation dot1q 10
;
;
   exit
;
   network ethernet0/0.20
; -- Ethernet Subinterface Configuration --
      ip vrf forwarding VRFE2
;
      ip address 192.168.212.201 255.255.254.0
;
;
;
      encapsulation dot1q 20
;
;
   exit
;
```

Example 1:

```
   feature access-lists
; -- Access Lists user configuration --
      access-list 100
         entry 1 default
         entry 1 permit
         entry 1 source address 192.168.212.201 255.255.255.255
;
      exit
;
   exit
;
   protocol ip
; -- Internet protocol user configuration --
      ipsec
; -- IPSec user configuration --
         enable
         assign-access-list 100
;
         template 1 isakmp des md5
         template 1 life duration seconds 1d
         template 1 keepalive dpd
;
         template 2 dynamic esp tdes sha1
         template 2 source-address 192.168.212.201
         template 2 encap transport
;
         map-template 100 2
```

In *Example 1*, all traffic from source 192.168.212.201 is protected by IPsec (regardless of the VRF it belongs to). That is:

- Packets that exit the router with source 192.168.212.201 are **encapsulated** in an IPsec tunnel, irrespective of whether they are sent through the ethernet0/0.10 or the ethernet0/0.20 subinterface.
- Packets that enter the router with destination 192.168.212.201 must be **encapsulated** in an IPsec tunnel or they will be dropped.

Example 2 is what we would get if we were to assign the template to a VRF:

Example 2:

```
   feature access-lists
; -- Access Lists user configuration --
      access-list 100
         entry 1 default
         entry 1 permit
         entry 1 source address 192.168.212.201 255.255.255.255
;
      exit
;
   exit
;
   protocol ip
; -- Internet protocol user configuration --
      ipsec
; -- IPSec user configuration --
         enable
         assign-access-list 100
;
         template 1 isakmp des md5
         template 1 life duration seconds 1d
         template 1 keepalive dpd
;
         template 2 dynamic esp tdes sha1
         template 2 source-address 192.168.212.201
         template 2 encap transport
         template 2 vrf VRF1
;
         map-template 100 2
```

In *Example 2,* only traffic with source 192.168.212.201 **transmitted through** VRF1 (i.e., **through an interface associated with VRF1**), is protected by IPsec. That is:

- Packets with source 192.168.212.201 that exit the router through **VRF1** (i.e., through the ethernet0/0 interface) are **encapsulated** in an IPsec tunnel.
- Packets with source 192.168.212.201 that exit the router via a VRF **other than** VRF1 (i.e., those that exit through the ethernet0/0.20 interface) do so in **plain**.
- Packets with destination 192.168.212.201 that enter the router through **VRF1** (i.e., through the ethernet0/0.10 interface) must be **encapsulated** in an IPsec tunnel or they will be dropped.
- Packets with destination 192.168.212.201 that enter the router via a VRF **other than** VRF1 (i.e., those entering through the ethernet0/0.20 interface) are **sent normally**, even if they haven't been encapsulated in IPsec.

Example 3 is what we would get if we were to add another dynamic template assigned to another VRF:

Example 3:

```
 feature access-lists
; -- Access Lists user configuration --
      access-list 100
         entry 1 default
         entry 1 permit
         entry 1 source address 192.168.212.201 255.255.255.255
;
      exit
;
   exit
;
   protocol ip
; -- Internet protocol user configuration --
      ipsec
; -- IPSec user configuration --
         enable
         assign-access-list 100
;
         template 1 isakmp des md5
         template 1 life duration seconds 1d
         template 1 keepalive dpd
```

```
;
        template 2 dynamic esp tdes sha1
        template 2 source-address 192.168.212.201
        template 2 encap transport
        template 2 vrf VRF1
;
        template 3 dynamic esp des md5
        template 3 source-address 192.168.212.201
        template 3 encap transport
        template 3 vrf VRF2
;
        map-template 100 2
        map-template 100 3
```

In *Example 3*, traffic with source address 192.168.212.201 **sent through VRF1** is protected with IPsec in **tdes** and **sha1** modes, while traffic with the same source address **sent through VRF2** is protected with IPsec in **des** and **md5** modes. That is:

- Packets with source address 192.168.212.201 that exit the router through **VRF1** (i.e., those exiting through the ethernet0/0.10 interface) are **encapsulated** in an IPsec tunnel in **tdes** and **sha1** modes.
- Packets with source address 192.168.212.201 that exit the router through **VRF2** (i.e., those exiting through the ethernet0/0.20 interface) are **encapsulated** in an IPsec tunnel in **des** and **md5** modes.
- Packets with destination address 192.168.212.201 that enter the router through **VRF1** (i.e., those entering through the ethernet0/0.10 interface) must be **encapsulated** in an IPsec tunnel in **tdes** and **sha1** modes, or they will be dropped.
- Packets with destination address 192.168.212.201 that enter the router through **VRF2** (i.e., those entering through the ethernet0/0.20 interface) must be **encapsulated** in an IPsec tunnel in **des** and **md5** modes, or they will be dropped.

Use the **no** form of this command to remove the template assignment.

“TEMPLATE [ID] FAULT-TOLERANT”

Enables fault tolerant IPsec recovery on the template. Any IPsec sessions set up using this template are automatically passed on to the fault-tolerant pair (making up the fault tolerant IPsec recovery system) when the router fails.

The available statistics are:

Example:

```
IPSec config>template 3 fault-tolerant
```

Use the **no** form of this command to stop the dynamic template participating in fault tolerant IPsec recovery.

“TEMPLATE [ID] REPORT {RX-LOSSES/RX-RATE-KBPS/TX-RATE-KBPS}”

Reports the configured IPsec statistics to the selected generic-input NSLA filter after a specified time period. For further information about NSLA filters, see the *bintec Dm754-I NSLA* manual.

The following statistics can be reported: RX-LOSSES, RX-RATE-KBPS and TX-RATE-KBPS.

Example:

```
IPSec config>template 3 report ?
  rx-losses       Report the percentage of lost packets
  rx-rate-kbps    Report the input rate (Kbps)
  tx-rate-kbps    Report the output rate (Kbps)
IPSec config>
```

“TEMPLATE [ID] REPORT RX-LOSSES NSLA-FILTER [FILTER ID] INTERVAL-MSEC [MSEC]”

Reports the packet loss percentage to the NSLA filter [FILTER-ID] every [MSEC] milliseconds.

Example:

```
IPSec config>template 3 report rx-losses ?
  nsla-filter    NSLA Filter that gets information
IPSec config>template 3 report rx-losses nsla-filter ?
  <1..65535>    Filter identifier
IPSec config>template 3 report rx-losses nsla-filter 1 ?
  interval-msec    Set report interval
IPSec config>template 3 report rx-losses nsla-filter 1 interval-msec ?
```

```
  <100..60000>    Time in milliseconds
IPSec config>template 3 report rx-losses nsla-filter 1 interval-msec 500
IPSec config>
```

Use the **no** form of this command to disable this function.

“TEMPLATE [ID] REPORT RX-RATE-KBPS NSLA-FILTER [FILTER ID] INTERVAL-SEC [SEC]”

Reports the template input rate (the sum of all the SAs INTO the template), in Kbps, to NSLA filter [FILTER-ID] every [SEC] seconds.

Example:

```
IPSec config>template 3 report rx-rate-kbps ?
  nsla-filter    NSLA Filter that gets information
IPSec config>template 3 report rx-rate-kbps nsla-filter ?
  <1..6553>    Filter identifier
IPSec config>template 3 report rx-rate-kbps nsla-filter 2 ?
  interval-msec    Set report interval
IPSec config>template 3 report rx-rate-kbps nsla-filter 2 interval-sec ?
  <1..3600>    Time in seconds
IPSec config>template 3 report rx-rate-kbps nsla-filter 2 interval-sec 1
IPSec config>
```

Use the **no** form of this command to disable this function.

“TEMPLATE [ID] REPORT TX-RATE-KBPS NSLA-FILTER [FILTER ID] INTERVAL-SEC [SEC]”

Reports the template output rate (the sum of all the SAs OUT of the template), in Kbps, to NSLA filter [FILTER-ID] every [SEC] seconds.

Example:

```
IPSec config>template 3 report tx-rate-kbps ?
  nsla-filter    NSLA Filter that gets information
IPSec config>template 3 report tx-rate-kbps nsla-filter ?
  <1..65535>    Filter identifier
IPSec config>template 3 report tx-rate-kbps nsla-filter 3 ?
  interval-msec    Set report interval
IPSec config>template 3 report tx-rate-kbps nsla-filter 3 interval-sec ?
  <1..3600>    Time in seconds
IPSec config>template 3 report tx-rate-kbps nsla-filter 3 interval-sec 1
IPSec config>
```

Use the **no** form of this command to disable this function.

Command history:

Release	Modification
11.00.05	Template **report** {RX-LOSSES/RX-RATE-KBPS/TX-RATE-KBPS} commands were introduced as of version 11.00.05.
11.01.00	Template **report** {RX-LOSSES/RX-RATE-KBPS/TX-RATE-KBPS} commands were introduced as of version 11.01.00.

2.3.2.2.6 ADVANCED command

The IPsec configuration main menu includes a command that allows you to configure a number of advanced characteristics in relation to the connection SAs created from dynamic templates. This command is called the *ADVANCED* command and provides access to several subcommands:

Command	Operation
DPD	Service for maintaining a SA connection.
KEEP-ALIVE	Service for maintaining a SA connection.
PURGE-TIMEOUT	Configuration for a SA timeout.
RENEGOTIATION-TIME	Service for performing SA re-negotiation.
EXPONENTIATION-DEVICE	Service for maintaining a SA connection.
LQUEUE	Cipher queue length.
NAT-T-PORT	UDP encapsulation port (NAT-T translation).
NAT-LOCAL-ADDRESS	Local addresses for rules that will be changed.

ADDRESS-ASSIGNED-TO-IFC	Interfaces that the IP addresses obtained during IKE negotiation take as destination.
PKT-DEST-ISAKMP-DEST	Packet destination indicates the remote tunnel address.
RRI-FLASH	Forces RRI-injected routes to be broadcast as quickly as possible through the dynamic routing algorithms.
CONNEVENT-PERIOD	Sets the periodic notification interval for open connections.
NUMBER-OF-IPSEC-HEADERS	Sets the maximum number of IPsec headers per packet.
NO	Sets the advanced IPsec configuration parameters to their default values.

“ADVANCED DPD”

Dead peer detection (DPD) is a service that allows the router to detect when the connection between the router and the device at the other end of the tunnel has been lost. To use the service, the router sends a DPD vendor ID during phase I of a negotiation. The router performs DPD verification by sending phase II notifications (R-U-THERE messages) to the device at the other end of the tunnel and waiting for DPD acknowledgements (R-U-THERE-ACK messages) to come back. The router will only send an R-U-THERE message if it doesn't receive traffic from the other end of the tunnel within a configurable period of time (idle period).

If you enable the service on an ISAKMP template, the router sends phase II DPD messages through the tunnels generated from that template, and also responds to notifications. If you don't enable the service, the router responds to any DPD notifications it receives but doesn't send any.

Command	**Operation**
ANTI-REPLAY	Enables the DPD packet anti-replay feature.
ALWAYS-SEND	Always sends keep-alive messages after the idle time has expired.
IDLE-PERIOD	Idle period before sending DPD packets.
INTERVAL	Interval between DPD keep-alive messages.
PACKETS	Maximum number of DPD packets without confirmation.
NO	Disables an option or resets a parameter's default values.

“ADVANCED DPD ANTI-REPLAY”

Enables the DPD packet anti-replay feature. This is disabled by default. Use the **no** form of this command to restore the default behavior.

“ADVANCED DPD ALWAYS-SEND”

Asks the router to perform DPD verification once the idle period has expired. When this is set to *ADVANCED DPD NO ALWAYS SEND* , the router must wait for data once the idle period has expired before performing DPD verification.

Command history:

Release	**Modification**
11.00.05	As of version 11.00.05, default is ADVANCED DPD ALWAYS-SEND.
11.01.00	As of version 11.01.00, default is ADVANCED DPD ALWAYS-SEND.

“ADVANCED DPD IDLE-PERIOD [SECONDS]”

Idle period (i.e., the time without receiving any data through the tunnel) before DPD verification is performed. The default value is 60 seconds. Use the **no** form of this command to restore the default value.

“ADVANCED DPD INTERVAL [SECONDS]”

Interval (in seconds) between sending DPD messages when there is no response. The default value is 5 seconds. Use the **no** form of this command to restore the default value.

“ADVANCED DPD PACKETS [MAX_PKTS]”

Maximum number of DPD messages without a response. Use the **no** form of this command to restore the 3-packet default value.

Example:

```
IPSec config>advanced dpd anti-replay
IPSec config>advanced dpd no always-send
IPSec config>advanced dpd idle-period 50
IPSec config>advanced dpd interval 4
```

```
IPSec config>advanced dpd packets 2
```

If you want the DPD service, you need to enable it individually on each ISAKMP template with the *TEMPLATE [ID] KEEPALIVE DPD* command.

“ADVANCED KEEP-ALIVE”

Keep alive is a service that tries to ensure the remote peer will maintain its SA open by monitoring the time during which the router shows no signs of life. When you enter this command, you will be asked to define two parameters:

Command	Operation
PACKETS	Maximum number of packets without a response.
TIMEOUT	Wait time, in seconds, since the last packet was received.
NO	Resets the above parameters to their default values.

Example:

```
IPSec config>advanced keep-alive packets 4
IPSec config>advanced keep-alive timeout 10
```

If you want the keep-alive service, enable it individually on each dynamic template using the *TEMPLATE ID KEEPALIVE KEEPALIVE* command.

“ADVANCED PURGE-TIMEOUT [SECONDS]”

Allows you to configure the SA timeout value. This is, for example, the time a SA negotiation will take to timeout when trying to negotiate a tunnel with a destination that isn't responding. Use the *ADVANCED NO PURGE-TIMEOUT* command to restore the 15-second default value for this parameter.

Example:

```
IPSec config>advanced purge-timeout ?
  <0s..3550w>    Value in the specified range
IPSec config>advanced purge-timeout 30s
```

“ADVANCED RENEGOTIATION-TIME”

The renegotiation time is a cap established in relation to the end of a SA connection lifetime that determines when a SA renegotiation should take place. If there is traffic between the renegotiation time limit and the end of the SA, the router will automatically renegotiate a new SA before the current SA lifetime expires. This stops traffic from getting lost when a SA expires.

This value is interpreted as a percentage. It is applied to each SA lifetime (in seconds) and never drops below one minute.

Use the **ADVANCED NO RENEGOTIATION-TIME** command to restore the 10% default value.

Example:

```
IPSec config>advanced renegotiation-time 20
Check-out time (%) - from SA's end-lifetime - to renegotiate : 20
```

The last line is used for confirmation purposes and describes the following behavior: when a SA has a twenty percent lifetime left, the router checks whether there are any packets up to the point of expiry. If there are, it initiates a new SA negotiation one minute before expiry.

“ADVANCED EXPONENTIATION-DEVICE”

Gives you access to two additional commands: HARDWARE and SOFTWARE. These allow you to configure the way ciphering operations will be carried out when processing encrypted packets. If you pick the HARDWARE option, packets will be encrypted using HARDWARE-level encryption (cipher card). With the SOFTWARE option, software code will be used to perform ciphering operations. Default is software.

Example:

```
IPSec config>advanced exponentiation-device ?
hardware      A hardware device will be used to carry out cipher operations
software      Software will be used to carry out cipher operations
IPSec config>advanced exponentiation-device hardware
```

“ADVANCED LQUEUE”

Configures the encryption queue length.

The use of fault tolerant IPsec recovery can lead to a large number of requests in the encryption queue. When using this subsystem, you must increase the value of this parameter to, at least, the number of IPsec sessions that will be established simultaneously, or the number of input buffers on the interface where the sessions are established, whichever is larger. Use the **no** form of this command to restore the default value (50).

Example:

```
IPSec config>advanced lqueue ?
  <0..65535>    Value in the specified range
IPSec config>advanced lqueue 25
```

“ADVANCED NAT-T-PORT”

Configures the port for UDP encapsulation (NAT-T translation).

The default value is 4500. Use the **no** form of this command to restore the default value.

Example:

```
IPSec config>advanced nat-t-port 10000
IPSec config>
```

“ADVANCED NAT-LOCAL-ADDRESS”

Sets the local address of rules exchanged through ISAKMP (please see the section on Configuring ISAKMP).

Example:

```
IPSec config>advanced nat-local-address ?
  <a.b.c.d>      Ipv4 format
  <interface>    Interface name
IPSec config>advanced nat-local-address ppp1
IPSec config>
```

Use the **no** form of this command to delete the local address.

“ADVANCED ADDRESS-ASSIGNED-TO-IFC”

Configures the interfaces with the IP addresses obtained through ISAKMP (please see the section on Configuring ISAKMP).

Example:

```
IPSec config>advanced address-assigned-to-ifc loopback1
IPSec config>
```

Use the **no** form of this command to delete the interfaces.

“ADVANCED PKT-DEST-ISAKMP-DEST”

Specifies that the IPsec remote tunnel endpoint address is the packet destination, thus inducing the tunnel to open.

Example:

```
IPSec config>advanced pkt-dest-isakmp-dest
```

Use the **no** form of this command to disable this function and restore the default value.

“ADVANCED RRI-FLASH”

When this command is enabled, RRI-injected routes are immediately broadcast through the dynamic routing algorithms (where possible) when they enter the device's routing table.

Example:

```
IPSec config>advanced rri-flash
IPSec config>
```

Use the **no** form of this command to stop the RRI-injected routes from being broadcast immediately. The dynamic routing algorithm in use will decide when to send them (default behavior).

“ADVANCED CONNEVENT-PERIOD”

You must enter a temporary value so that a **CONNEVENT** event (which will use the value as a time interval) can take place at regular intervals. This type of event is generated for each connection established, with between 40 and 50 events generated per second. The router will seek to concentrate the events at the beginning of each interval, but the

process will take longer if many tunnels are open. The interval that you enter must be long enough to allow all of the open connections to be notified. If it isn't, you can use the **list statistics** command from the monitoring console to view the number of connections that couldn't be notified.

Example:

```
IPSec config>advanced connevent-period ?
  <0s..3550w>    Value in the specified range
IPSec config>advanced connevent-period 5m
IPSec config>
```

Use the **no** form of this command to disable periodic notifications (which uses the CONNEVENT event) and restore the period value to its 0 second default.

“ADVANCED NUMBER-OF-IPSEC-HEADERS”

Establishes the maximum number of IPsec headers a packet can have. That is, the number of successive IPsec encapsulations that can be done. The default value is 1.

Example:

```
IPSec config>advanced number-of-ipsec-headers 2
IPSec config>
```

Use the **no** form of this command to restore the default value.

2.3.2.2.7 KEY PRESHARED command

This concludes the configuration of the ISAKMP/IKEv2 and dynamic templates required to perform IPsec IKE. Of course, we need to enter one more parameter to make them work. That parameter is the pre-shared key that both security routers must possess in order to authenticate one another. The key is entered from the main IPsec menu:

“KEY PRESHARED {IP/HOSTNAME} [ADDRESS/NAME] {CIPHERED/PLAIN} [KEY]”

Pre-shared key authentication works differently for IKEv1 and IKEv2 negotiations:

- In the case of IKEv1:

  You can enter the pre-shared key associated with the remote device name or IP address, according to how the tunnel was envisaged when the *TEMPLATE IKE IDTYPE* command was used for the ISAKMP templates.
- In the case of IKEv2:

  When acting as the IKEv2 initiator, you can enter the pre-shared key associated with the responder's hostname or IP address. The pre-shared key lookup is performed using the peer's hostname or address, in that order.

  When acting as the IKEv2 responder, you can enter the pre-shared key associated with the Initiator ID (name or IP), according to how the tunnel was envisaged when the *TEMPLATE IKE IDTYPE* command was used for the IKEv2 templates on the initiator.

Please note, however, that this command doesn't need an identifier [ID] like other commands because it is associated with a remote IP address/host rather than with a template.

You can enter the pre-shared key in plain text (using the *PLAIN* subcommand), or in ciphered text (using the *CIPHERED* subcommand). The key is usually entered in plain when entered manually from the console. When using a configuration saved in text mode (from a *SHOW CONFIG* command), the key will be ciphered. Plain text keys can be between 1 and 32 bytes long and can be entered in hexadecimal format (starting with 0x), or in ASCII. Don't forget to double the number of characters (between 0-9 and A-F) when you enter a hexadecimal value. Ciphered keys are always shown in hexadecimal format.

Example 1:

```
IPSec config>key preshared ip 192.100.1.1 plain 1234567890
IPSec config>key preshared hostname Router2 plain 1234567890bintec
IPSec config>key preshared ip 192.100.1.1 plain 0x1234567890abcdef
```

The pre-shared key admits networks with 0, 8, 16 and 24-bit masks in IP addresses.

Example 2:

```
IPSec config>key preshared ip 192.100.1.0 plain 1234567890
```

This key is assigned to the entire 192.100.1.0 255.255.255.0 network.

The pre-shared key admits the wildcard character (*) at the end of the host name.

Example 3:

```
IPSec config>key preshared hostname Router* plain 1234567890bintec
```

This key is assigned to Router1, Router, Router_234...

The most restrictive intersection (if there are any) will always be taken.

Example 4:

```
IPSec config>key preshared hostname Router* plain 1234567890bintec
IPSec config>key preshared hostname Router plain 1111111
```

Key 1111111 will be used if the hostname is *Router*.

```
IPSec config>key preshared ip 192.100.1.0 plain 1234567890
IPSec config>key preshared ip 192.100.1.163 plain aaaa
```

Key aaaa will be used if the IP is 192.100.1.163.

Use the *LIST KEY PRESHARED* command to view the configured pre-shared keys. The keys as such are not printed in the console but you can see which IP addresses or host names are associated with a pre-shared key:

```
IPSec config>list key preshared
5 key entries
   192.100.1.1 ************
   Router2 ************
   192.100.1.0 ************
   Router* ************
   Router ************
```

You can use the *NO KEY PRESHARED {IP/HOSTNAME} [ADDRESS/NAME]* command to remove a key associated with an IP address or host name.

```
IPSec config>no key preshared ip 192.100.1.0
```

2.3.2.3 Creating the SPD

Finally, having defined the access control list (ACL) and templates, all we need to do now is to create a security policy database (SPD). Each entry in the SPD consists of an ACL element and an associated template. Mapping is the term given to associating the ACL element with the template. See below to learn about the **map** command and how to use it:

Command	Operation
ASSIGN-ACCESS-LIST	Assigns an ACL to the IPsec protocol.
ASSOCIATE-KEY	Associates a key with an ACL.
ASSOCIATE-DEST-MASK	Associates a destination mask with an ACL.
MAP-TEMPLATE	Associates ACL elements with templates.

“ASSIGN-ACCESS-LIST [ACCESS_LIST]”

Assigns an extended ACL to the IPsec protocol.

Example:

```
IPSec config>assign-access-list ?
  <100..1999>    Enter extended access list id
IPSec config>assign-access-list 100
```

To remove the ACL assignment, use the **no** form of this command.

“ASSOCIATE-KEY {IP/HOSTNAME} [ACCESS_LIST] [ADDRESS/NAME KEY]”

One of the parameters negotiated when opening an IPsec tunnel is the access control, i.e., the *tunnel clients*. In principle, knowing the pre-shared key allows the remote device to open a tunnel to the local device, independently of the clients. However, sometimes this is not convenient, as you may want to allow the devices that know one key to have different controls to the devices that know another key.

We can make the following statements with regard to *Example 1* below:

Only devices that know the key associated with the *bintec_router* hostname can open a tunnel to access the 192.60.64.0/24 network.

Devices that only know the key associated with the *router* hostname **cannot** open a tunnel to access the entire 192.60.64.0/24 network.

Since there is no key associated with ACL 101, devices that know the keys associated with the *router* and *bintec_router* hostnames can open a tunnel to access host 192.60.64.1

Example 1:

```
Extended Access List 101, assigned to IPSec

1     PERMIT  SRC=192.60.64.1/32  DES=0.0.0.0/16  Conn:0

Extended Access List 100, assigned to IPSec

10    PERMIT  SRC=192.60.64.0/24  DES=0.0.0.0/16  Conn:0
IPSec config>list key preshared
2 key entries
   bintec_router ************
   router ************
IPSec config>associate-key hostname 100 bintec_router
```

We can make the following statements with regard to *Example 2* below:

- The devices that know the key associated with an IP address starting with 10 are the only ones allowed to open a tunnel for production. In this case, a production tunnel can only be opened for devices that know the key for IP address 10.127.0.28. The *ASSOCIATE-KEY IP 100 10.0.0.0* command controls this behavior.
- All devices that know the key associated with an IP address starting with 10, or the generic key (entered with *KEY PRESHARED IP 0.0.0.0*), can open a management tunnel with the *10.127.1.57* management device.

Example 2:

```
feature access-lists
; -- Access Lists user configuration --
   access-list 100
      description "Control de Acceso para Produccion"
      entry 1 default
      entry 1 permit
      entry 1 source address 10.127.1.0 255.255.255.0
      entry 1 destination address 10.127.0.0 255.255.0.0
;
   exit
;
   access-list 101
      description "Control de Acceso para Gestion"
;
      entry 1 default
      entry 1 permit
      entry 1 source address 10.127.1.57 255.255.255.255
      entry 1 destination address 10.127.0.0 255.255.0.0
;
   exit
;
exit
p ip
; (...)
   ipsec
; -- IPSec user configuration --
      enable
      assign-access-list 101
      assign-access-list 100
;
; (...)
      associate-key ip 100 10.0.0.0
      key preshared ip 0.0.0.0 ciphered 0x37349246263B0066
      key preshared ip 10.127.0.28 ciphered 0x7CC9756395EFB97F
   exit
```

Use the **no** form of this command to remove an association between a key and an ACL.

“ASSOCIATE-DEST-MASK [ACCESS_LIST] [MASK]”

One of the parameters negotiated when opening an IPsec tunnel is the access control, i.e., the *tunnel clients*. In a star configuration, for example, tunnel clients are usually allocated different subnets of a given network. Consider, for instance, a network where the remote ends are assigned class C IP addresses from network 192.168.0.0/16. Then, the remote end *A* would have the 192.168.1.0/24 address range, the remote endpoint *B* would have the 192.168.2.0/24 address range, and so on.

In this case, you would only need to configure one ACL with 192.168.0.0/16 as destination in the central remote. The problem with this configuration is that it would not prevent the remote ends from opening a tunnel to a wider range of addresses than desired (i.e., 255 addresses). By setting the **associate-dest-mask** parameter to 255.255.255.0, you can set any limit you want. As such, this parameter gives you a very simple configuration on the central side, protected from negotiations attempting to propose a wider address range than permitted, without having to configure a different access list for each one.

We can make the following statements with regard to the example below:

- Remote devices are only allowed to open a tunnel with a subset of the 192.168.0.0/16 network and this subset has a mask with more bits set to 1 than 255.255.255.0, which is what is configured in the **associate-dest-mask** parameter.
- Any remote device attempting to open a tunnel with a mask with fewer bits set to 1 than configured in the **associate-dest-mask** parameter will have its proposal rejected and will receive an *invalid-id* message.

Example:

```
Extended Access List 101, assigned to IPSec

1     PERMIT  SRC=192.60.64.1/32  DES=192.168.0.0/16  Conn:0

IPSec config>associate-dest-mask 101 255.255.255.0
```

Use the **no** form of this command to remove an association between a destination mask and an ACL.

“MAP-TEMPLATE [ACCESS_LIST] [Template ID]”

Associates an ACL element with a template to create an SPD element.

Example:

```
IPSec config>map-template 100 4
```

When performing mapping, the ACL entry list from the IPsec monitoring menu will sometimes show some automatically generated entries distinguished by the words *DYNAMIC ENTRY* that weren't entered by the user. They are important in that they allow the two tunnel ends to communicate control packets.

```
Extended Access List 101, assigned to IPSec

 ACCESS LIST ENTRIES
0     DENY    SRC=0.0.0.0/0  DES=192.60.64.1/32  Conn:0
        PROT=17  SPORT=500
        DYNAMIC ENTRY
        Hits: 0

0     DENY    SRC=0.0.0.0/0  DES=192.60.64.1/32  Conn:0
        PROT=17  SPORT=4500
        DYNAMIC ENTRY
        Hits: 0

0     DENY    SRC=0.0.0.0/0  DES=192.60.64.1/32  Conn:0
        PROT=50-51
        DYNAMIC ENTRY
        Hits: 0

0     PERMIT  SRC=192.60.64.2/32  DES=192.60.64.1/32  Conn:0
        DYNAMIC ENTRY
        Hits: 0

1     PERMIT  SRC=0.0.0.6/32  DES=192.60.64.1/32  Conn:0
        Hits: 0
```

Use the **no** form of this command to remove an association (or mapping) between an ACL element and a template. The automatically generated entry remains even if you disable mapping (i.e., you should delete it if it is no longer re-

quired).

Mapping is the final step when configuring IPsec.

However, there is an additional command related to the bandwidth reservation feature (BRS):

“QOS-PRE-CLASSIFY”

Allows you to classify packets based on their respective BRS classes prior to encryption.

When this mode is enabled, packets are classified prior to encryption, which means that different traffic classes can be prioritized within the same IPsec tunnel. Classification will not work on access controls that aren't associated with an IP rule. This is because there is no way of knowing which outbound interface the packets will use before being encrypted and, therefore, no way of applying the BRS associated with that interface. When this mode is disabled, the analyzed header becomes the IPsec tunnel header and all traffic coming from the IPsec tunnel is then classified in the same BRS class.

Example:

```
IPSec config>qos-pre-classify
IPSec config>
```

Run the *NO QOS-PRE-CLASSIFY* command to disable this option and set the default behavior.

Before considering the configuration complete, you can use trace monitoring to check what you've done, modify errors and even determine which events to view:

Command	Operation
LIST ALL	Displays the entire configuration.
SHOW CONFIG	Displays the configuration commands.
EVENT	Enables certain events.
LIST ENABLED-EVENTS	Displays the filter configured for events monitoring (if any).

“LIST ALL”

Shows all the configuration policies held in the SPD (i.e., the ACL elements and the template list).

Example:

```
IPSec config>list all
IPSec Access Control.
Access Control is: enabled
QOS Preclassify is: disabled

Access Lists assigned to IPSec:
  Extended Access List 101
  Templates: 1

Extended Access List 101, assigned to IPSec

1     PERMIT  SRC=0.0.0.6/32  DES=192.60.64.1/32  Conn:0

TEMPLATES
1 dynamic ESP-3DES ESP-MD5  SRC=0.0.0.6 DES=192.60.64.1
   LifeTime:0h3m0s 100000 kbytes
   PFS disabled

2 dynamic ESP-DES ESP-SHA1  SRC=192.24.51.75 DES=192.24.51.74
   LifeTime:0h50m0s 100000 kbytes
   PFS disabled

3 dynamic AH-MD5  SRC=192.24.51.75 DES=192.24.51.74
   LifeTime:0h50m0s 100000 kbytes
   PFS disabled

4 dynamic AH-SHA1  SRC=192.24.51.75 DES=192.24.51.74
   LifeTime:0h50m0s 100000 kbytes
   PFS disabled

20 isakmp 3DES MD5  DES=192.60.64.1
   LifeTime:0h4m0s
```

```
   IKE AGGRESSIVE
   PRESHARED
   fqdn ID TYPE
   OAKLEY GROUP 1

4 key entries
   172.24.51.57 ************
   192.24.51.74 ************
   192.24.78.75 ************
   192.60.64.1 ************
0 rsakey entries
Id.           Date.         Len         CA.                   Cert sn.

KeepAlive Configuration:
   Maximum number of encoded packets without receiving an answer: 0.
   Timeout after last packet encoded: 0 seconds.

DPD Configuration:
Idle period(secs) before sending DPD keepalives: 60
Maximum number of DPD keepalives not acknowledged: 3
Period of time(secs) between DPD keepalives: 5
Always send keepalive after idle period expiration : ENABLED
Anti-replay : DISABLED

Check-out time (%) - from SA's end-lifetime - to renegotiate : 10

SA's purge timeout:  15

Use software exponentiation
IPSec config>
```

“SHOW CONFIG”

Displays the configuration commands. Please note that any field values matching default values will not be shown. The following example displays the *SHOW CONFIG* command output with the same configuration as the *LIST ALL* command example.

Example:

```
IPSec config>show config
; Showing Menu and Submenus Configuration for access-level 15 ...

      enable
      assign-access-list 101
;
      template 1 dynamic esp tdes md5
      template 1 source-address 0.0.0.6
      template 1 destination-address 192.60.64.1
      template 1 life type both
      template 1 life duration seconds 180
      template 1 life duration kbytes 100000
;
      template 2 dynamic esp des sha1
      template 2 source-address 192.24.51.75
      template 2 destination-address 192.24.51.74
      template 2 life type both
      template 2 life duration seconds 3000
      template 2 life duration kbytes 100000
;
      template 3 dynamic ah md5
      template 3 source-address 192.24.51.75
      template 3 destination-address 192.24.51.74
      template 3 life type both
      template 3 life duration seconds 3000
      template 3 life duration kbytes 100000
;
      template 4 dynamic ah sha1
      template 4 source-address 192.24.51.75
```

```
      template 4 destination-address 192.24.51.74
      template 4 life type both
      template 4 life duration seconds 3000
      template 4 life duration kbytes 100000
;
      template 20 isakmp tdes md5
      template 20 destination-address 192.60.64.1
      template 20 life duration seconds 240
      template 20 ike ca THAWTECA.CER
      template 20 ike mode aggressive
      template 20 ike idtype fqdn
;
      map-template 101 1
      key preshared ip 172.24.51.57 hello
      key preshared ip 192.24.51.74 ciphered 0xF85C0CB62556C562120794C28EB9334
      key preshared ip 192.24.78.75 ciphered 0xF85C0CB62556C562120794C28EB9334
      key preshared ip 192.60.64.1 ciphered 0xF85C0CB62556C562120794C28EB9334
IPSec config>
```

“EVENT ALL”

Lets you view all events. Events must be enabled through the events monitoring process (P 3) and can be viewed in P 2.

Example:

```
IPSec config>event all
```

“EVENT ADDRESS-FILTER [IP ADD][MASK]”

You can use this command to view only those events with a source or destination address within the range defined by [IP ADD][MASK], once enabled. Please see the *LIST NEGOTIATION FILTER* monitoring command.

Example:

```
IPSec config>event address-filter 192.100.1.2 255.255.255.255
```

“LIST ENABLED-EVENTS”

Displays the filter configured for event monitoring (if any).

Example:

```
IPSec config>list enabled-events
 Address/Subnet enabled : 192.100.1.2 with MASK : 255.255.255.255
```

2.3.3 ISAKMP configuration mode

The ISAKMP configuration mode provides us with a method to configure the phase II parameters that will be negotiated once the phase I negotiation is complete. With this method we can define, in a secure manner, what the characteristics of the IPsec session negotiated in phase II for data exchange will be. At the time this document was written, the properties and operating mode for this configuration mode can be found in the draft entitled *ISAKMP Configuration Mode*.

This method is usually used in star networks. In such networks, the devices that want to connect to the VPN are allocated an address for the session by a central node. This node is also responsible for determining which devices will be name servers and whether PFS or the port will be used for NAT traversal (NAT-T).

Use the following parameters from the TEMPLATE menu to configure this method:

Command	Operation
IKE METHOD	Adds *XAUTH-INIT-PRESHARED* and *XAUTH-INIT-RSA* options.
CONFIG	Allows you to specify whether the device will initiate the configuration method, wait for a proposal, or behave in accordance with the IKE method used.

“IKE METHOD XAUTH-INIT-PRESHARED”

Lets you add the new *Extended Pre-shared Key Authentication* functionality to the above-mentioned *IKE METHOD* command. At the time this document was written, this functionality is described in the *Extended Authentication within ISAKMP/Oakley* draft. If you enable this parameter, it means that you want to perform pre-shared authentication with an *ISAKMP configuration* process, whereby the initiator must authenticate with a remote server that can allocate the initiator with the internal VPN IP address, among other things.

Example:

```
IPSec config>template 4 ike method xauth-init-preshared
```

“IKE METHOD XAUTH-INIT-RSA”

Lets you add the new *Extended RSA Authentication* functionality to the *IKE METHOD* command. At the time this document was written, this functionality is described in the *Extended Authentication within ISAKMP/Oakley* draft. If you enable this parameter, it means that you want to perform RSA authentication with an *ISAKMP configuration* process, whereby the initiator must authenticate with a remote server that can allocate the initiator with the internal VPN IP address, among other things.

Example:

```
IPSec config>template 4 ike method xauth-init-rsa
```

“CONFIG INITIATOR”

Specifies that the router will initiate configuration, making the initial proposals and requesting the necessary parameters.

Example:

```
IPSec config>template 4 config initiator
```

“CONFIG RESPONDER”

Specifies that the router will wait for the remote device to initiate configuration.

Example:

```
IPSec config>template 4 config responder
```

“CONFIG NONE”

Specifies that the IKE method used will determine whether the router acts as initiator or responder. This is the default behavior.

Example:

```
IPSec config>template 4 config none
```

2.3.3.1 EXTENDED AUTHENTICATION

With Extended Authentication (XAUTH), the router authenticates itself with a server and is subsequently assigned the parameters required to establish a connection. This authentication is usually done by means of a user name and password.

The following commands allow you to associate a user with a password, IP address or name.

Command	Operation
XAUTH-IP	Associates a user with an IP address.
XAUTH-HOSTNAME	Associates a user with a name.
XAUTH-USER	Specifies a user’s properties.

“XAUTH-IP [IP address] USER [user name]”

“XAUTH-IP [IP address] PASSWORD [password]”

“XAUTH-IP [IP address] LOCAL-LAN-ACCESS [network]”

“XAUTH-IP [IP address] MASK [mask]”

“XAUTH-IP [IP address] DNS-SERVER [primary-server] [secondary-server]”

“XAUTH-IP [IP address] DEFAULT-DOMAIN-NAME [domain-name]”

“XAUTH-IP [IP address] NO REQUEST IP-ADDRESS”

Using these commands you can define the username and password that will be associated with the IP address parameter you enter.

In the case of the XAUTH responder (i.e., when the *CONFIG RESPONDER* command is configured), the IP address will be the address that was used to identify the remote endpoint. Under normal procedure, the XAUTH responder requests an IP address upon completion of the phase phase I negotiations and authentication. However, if the *NO RE-*

QUEST IP-ADDRESS option is configured, the XAUTH responder proceeds directly to phase II negotiations after obtaining authentication, using the *clients* configured in the associated ACL.

In the case of the XAUTH initiator (i.e., when the *CONFIG INITIATOR* command is configured), this IP address will be the address assigned to the remote endpoint (i.e., to the XAUTH responder) during negotiation of the ISAKMP configuration method (if you want to get the address from a pool, you should use the *XAUTH-USER* command rather than this command). The *MASK* command allows you to change the 32-bit network mask default that the XAUTH initiator sends to the XAUTH responder. If a *MASK* is already configured in the responder, it overwrites the value received from the initiator. If you want the DNS servers and the default domain name that the XAUTH initiator sends to the XAUTH responder to be different from the ones that the XAUTH initiator uses, you can use the *DNS-SERVER* and *DEFAULT-DOMAIN-NAME* options to change them. You can also use the *LOCAL-LAN-ACCESS* command to specify that the XAUTH responder's local network remain outside the IPsec tunnel. That is, the network won't be protected by IPsec on the XAUTH responder side. This command is often used to allow the XAUTH responder to access its local network (usually a restricted access LAN) while bypassing IPsec policies.

Example:

```
IPSec config>xauth-ip 1.1.1.1 user router1
IPSec config>xauth-ip 1.1.1.1 password plain mykey
IPSec config>xauth-ip 1.1.1.1 local-lan-access 192.168.1.0 255.255.255.0
```

“XAUTH-HOSTNAME [hostname] USER [user name]”

“XAUTH-HOSTNAME [hostname] PASSWORD [password]”

These two commands let you define the user and password that will be associated with the name entered as a parameter.

This name indicates the host name used to identify the remote endpoint.

Example:

```
IPSec config>xauth-hostname remoterouter user router1
IPSec config>xauth-hostname remoterouter password plain mykey
```

“XAUTH-USER [user] POOL [pool name]”

“XAUTH-USER [user] PASSWORD [password]”

“XAUTH-USER [user] LOCAL-LAN-ACCESS [network]”

“XAUTH-USER [user] MASK [mask]”

“XAUTH-USER [user] DNS-SERVER [primary-server] [secondary-server]”

“XAUTH-USER [user] DEFAULT-DOMAIN-NAME [domain-name]”

“XAUTH-USER [user] NO REQUEST IP-ADDRESS”

These commands allow you to associate a user with a pool, password, local network outside the tunnel and network mask.

In the case of the XAUTH initiator, when the user (XAUTH responder) is identified with their user name, *user,* and then requests an IP address, the address will come from the pool configured as *pool name* (this pool will have been previously defined in the IP configuration). Note, if you want the IP address to remain the same at all times, then you should use the *XAUTH-IP* command rather than this command. The *MASK* command, like the *XAUTH-IP* command, allows you to change the 32-bit network mask default that the XAUTH initiator sends to the XAUTH responder. If a *MASK* is already configured in the responder, it overwrites the value received from the initiator. The *DNS-SERVER* and *DEFAULT-DOMAIN-NAME* options allow you to change the DNS servers and default domain name that the XAUTH initiator sends to the XAUTH responder so that they aren't the same as the ones used by the XAUTH initiator. In addition, the *LOCAL-LAN-ACCESS* like the *XAUTH-IP* command, specifies the XAUTH responder’s local network which is outside the IPsec tunnel.

Example:

```
IP config>pool remotevpn 172.24.100.80 172.24.100.95
IPSec config>xauth-user myuser pool remotevpn
IPSec config>xauth-user myuser mask 255.255.255.0
IPSec config>xauth-user myuser password plain mykey
IPSec config>xauth-user myuser local-lan-access 192.168.1.0 255.255.255.0
```

In the above example, the dialogue follows this sequence:

(1) When phase I negotiations conclude, the XAUTH initiator requests the user ID and password.

(2) The XAUTH responder sends its user name and password.

(3) The XAUTH initiator checks they are correct and returns a positive response.

(4) The XAUTH responder then requests an IP address, mask, DNS server, default domain name, local access network, NAT port, and other parameters.

(5) The XAUTH initiator gets an IP address from the remote VPN pool and sends it to the XAUTH responder along with the mask, local access network, DNS server, default domain name, NAT port and any other requested parameters.

(6) The XAUTH responder begins phase II negotiations using the assigned IP/mask.

Please note that steps 4 and 5 don't appear when the *NO REQUEST IP-ADDRESS* command is configured in the XAUTH responder. In this case, the XAUTH responder goes straight to the phase II negotiations after obtaining authentication, using the clients configured in the associated access list.

Command history:

Release	Modification
10.09.26	The *MASK*, *DNS-SERVER* and *DEFAULT-DOMAIN-NAME* options were introduced as of version 10.09.26.
11.00.05	The *XAUTH-IP*, *XAUTH-HOSTNAME* and *XAUTH-USER* default commands are obsolete as of version 11.00.05. The *MASK*, *DNS-SERVER* and *DEFAULT-DOMAIN-NAME* options were introduced as of version 11.05.00.
11.01.00	The *XAUTH-IP*, *XAUTH-HOSTNAME* and *XAUTH-USER* default commands are obsolete as of version 11.01.00. The *MASK* option was introduced as of version 11.01.00.
11.01.01	The *DNS-SERVER* and *DEFAULT-DOMAIN-NAME* options were introduced as of version 11.01.01.
10.09.27	The *MASK* value configured on the responder overwrites the value received from the initiator as of version 10.09.27.
11.00.06	The *MASK* value configured on the responder overwrites the value received from the initiator as of version 11.00.06.
11.01.02	The *MASK* value configured on the responder overwrites the value received from the initiator as of version 11.01.02.

2.3.3.2 Configuration example: Router server for VPN clients

Imagine we have a router that closes VPN client tunnel connections. The router has an Ethernet interface and an ADSL interface.

The Ethernet network is 172.24.0.0/16 and the address is 172.24.78.130.

The ADSL address is 80.1.1.123.

```
   network ethernet0/0
; -- Ethernet Interface User Configuration --
      ip address 172.24.78.130 255.255.0.0
      ip proxy-arp enable
   exit
;
   network atm0/0.1
; -- ATM interface configuration --
      ip address 80.1.1.123 255.255.255.255
   exit
```

The device has subnet 172.24.6.80 255.255.255.240 available for users connecting remotely. This subnet is configured in the *remotevpn* pool.

```
   feature access-lists
; -- Access Lists user configuration --
      access-list 101
;
      entry 1 default
      entry 1 permit
      entry 1 destination address 172.24.6.80 255.255.255.240
;
      exit
   exit
;
   protocol ip
; -- Internet protocol user configuration --
```

```
      route 172.24.6.80 255.255.255.240 80.1.1.123
      pool remotevpn 172.24.6.80 172.24.6.95
;
   exit
```

The configuration for user *daisy* would be:

```
   protocol ip
      ipsec
         xauth-user daisy pool remotevpn
         xauth-user daisy password plain goodbye
```

The group is called *mygroup* and the key would be *hello*.

```
   protocol ip
      ipsec
         key preshared hostname mygroup plain hello
```

Complete configuration

```
   log-command-errors
   no configuration
   add device atm-subinterface atm0/0 1
   feature access-lists
; -- Access Lists user configuration --
      access-list 101
         entry 1 default
         entry 1 permit
         entry 1 destination address 172.24.6.80 255.255.255.240
;
      exit
;
   exit
;
;
   network ethernet0/0
; -- Ethernet Interface User Configuration --
      ip address 172.24.78.130 255.255.0.0
      ip proxy-arp enable
   exit
;
   network atm0/0.1
; -- ATM interface configuration --
      ip address 80.1.1.123 255.255.255.255
   exit
;
;
   protocol ip
; -- Internet protocol user configuration --
      route 0.0.0.0 0.0.0.0 80.1.1.123 1
      route 172.24.6.80 255.255.255.240 80.1.1.123
;
      pool remotevpn 172.24.6.80 172.24.6.95
;
      rule 1 local-ip 80.1.1.123 remote-ip any
      rule 1 napt translation
      rule 1 napt firewall
      rule 1 napt timeout 30
;
      classless
;
;
      ipsec
; -- IPSec user configuration --
         enable
         assign-access-list 101
;
         template 1 isakmp tdes md5
         template 1 life duration seconds 86400
```

```
         template 1 ike mode aggressive
         template 1 ike method xauth-init-preshared
         template 1 ike group two
         template 1 keepalive dpd
;
         template 2 dynamic esp tdes md5
         template 2 source-address 80.1.1.123
;
         map-template 101 2
         key preshared hostname mygroup plain hello
         advanced purge-timeout 30
;
         xauth-user daisy pool remotevpn
         xauth-user daisy password plain goodbye
;
;
      exit
;
   exit
;
   feature dns
; -- DNS resolver user configuration --
      server 172.24.0.7
   exit
;
   dump-command-errors
   end
```

2.3.3.3 VPN client configuration when the client is a router and doesn't require an IP address

In this example, the router is configured as a client of the server from the above example and doesn't request an IP address.

```
   log-command-errors
   no configuration
   set hostname mygroup
   feature access-lists
; -- Access Lists user configuration --
      access-list 100
         entry 1 default
         entry 1 permit
         entry 1 source address 172.24.6.84 255.255.255.252
;
      exit
;
   exit
;
;
   network ethernet0/0
; -- Ethernet Interface User Configuration --
      ip address 192.168.248.28 255.255.255.0
      ip address 172.24.6.85 255.255.255.252 secondary
;
   exit
;
;
   event
; -- ELS Config --
      enable trace subsystem IKE ALL
   exit
;
;
   protocol ip
; -- Internet protocol user configuration --
      route 0.0.0.0 0.0.0.0 192.168.248.98
;
;
```

```
      ipsec
; -- IPSec user configuration --
         enable
         assign-access-list 100
;
         template 1 isakmp tdes md5
         template 1 destination-address 80.1.1.123
         template 1 ike mode aggressive
         template 1 ike natt-version draft-v2-n
         template 1 config responder
         template 1 ike method xauth-init-preshared
         template 1 ike idtype keyid
         template 1 ike group two
;
         template 2 dynamic esp tdes md5
         template 2 source-address 192.168.248.28
         template 2 destination-address 80.1.1.123
;
         map-template 100 2
         key preshared ip 80.1.1.123 plain hello
;
;
         xauth-ip 80.1.1.123 user daisy
         xauth-ip 80.1.1.123 password plain goodbye
         xauth-ip 80.1.1.123 no request ip-address
;
      exit
;
   exit
;
   dump-command-errors
   end
```

2.3.3.4 ASSIGNED IP ADDRESS DESTINATION

The device behaving as a client can receive an IP address during the ISAKMP configuration protocol. This IP address can be used in two different ways:

- As a NAT address in the NAPT rules.
- As an interface address.

As a NAT address in the NAPT rules

The assigned IP address becomes the NAT address used in the NAPT rules whose local address or interface matches the address configured with the *ADVANCED NAT-LOCAL-ADDRESS* command.

In this operating mode, you need to know the network the server-assigned IP address belongs to. You'll also need to use a fictitious address from that network in the NAPT rule in order to trigger the ISAKMP negotiation when there is traffic.

In the sample configuration below, the server assigns an IP address from network 172.24.0.0/16 and uses one of the network's addresses (172.24.78.1) as the IP address in the NAPT rule. It doesn't matter which network IP address is used in the rule because it is changed during the ISAKMP exchange and is never used.

```
   log-command-errors
   no configuration
   set hostname bintec
   add device ppp 1
   set data-link at cellular0/0
   set data-link at cellular0/1
   set data-link at cellular1/0
   set data-link at cellular1/1
   feature access-lists
; -- Access Lists user configuration --
      access-list 100
         entry 1 default
         entry 1 permit
         entry 1 source address 172.24.0.0 255.255.0.0
      exit
```

```
   exit
;
   global-profiles dial
; -- Dial Profiles Configuration --
      profile HSPA default
      profile HSPA inout
      profile HSPA 3gpp-apn movistar.es
;
      profile MOVISTAR default
      profile MOVISTAR dialout
      profile MOVISTAR idle-time 300
   exit
;
   network ethernet0/0
; -- Ethernet Interface User Configuration --
      ip address 192.6.3.186 255.255.255.0
;
   exit
;
   network cellular1/0
; -- Interface AT. Configuration  --
      pin plain 1111
      sim-select internal-socket-2
;
      network mode automatic
      network domain cs+ps
   exit
;
   network cellular1/1
; -- Interface AT. Configuration  --
      lcp-options acfc
      lcp-options pfc
      lcp-options accm 0
exit
;
   network ppp1
; -- Generic PPP User Configuration --
      ip address unnumbered
;
      ppp
; -- PPP Configuration --
         authentication sent-user MOVISTAR password keykey
         ipcp local address assigned
         no ipcp peer-route
         lcp echo-req off
      exit
;
      base-interface
; -- Base Interface Configuration --
         base-interface cellular1/1 link
         base-interface cellular1/1 profile HSPA
;
      exit
;
   exit
;
   network loopback1
; -- Loopback interface configuration --
      ip address unnumbered
   exit
;
   event
; -- ELS Config --
      enable trace subsystem IKE ALL
   exit
;
;
```

```
   protocol ip
; -- Internet protocol user configuration --
      route 0.0.0.0 0.0.0.0 ppp1
;
      rule 1 local-ip ppp1 remote-ip any
      rule 1 napt translation
      rule 1 napt ip 172.24.78.1
;
      classless
;
      ipsec
; -- IPSec user configuration --
         enable
         assign-access-list 100
;
         template 1 isakmp tdes md5
         template 1 destination-address 80.36.189.231
         template 1 ike mode aggressive
         template 1 ike natt-version draft-v3
         template 1 config responder
         template 1 ike method xauth-init-preshared
         template 1 ike idtype keyid
         template 1 ike group two
;
         template 2 dynamic esp tdes md5
         template 2 source-address ppp1
         template 2 destination-address 80.36.189.231
;
         map-template 100 2
         key preshared ip 80.36.189.231 plain key1
         advanced nat-local-address ppp1

;
         xauth-ip 80.36.189.231 user anonymous
         xauth-ip 80.36.189.231 password plain pppp
;
      exit
;
   exit
   dump-command-errors
   end
```

Notes:

- All traffic is protected by IPsec, and has the source address assigned by the server.
- You can't use the NAPT rule firewall command in this configuration.
- The device does not reply to the assigned IP address. That is, it cannot be managed (telnet, ftp, snmp, etc.) from the server network.
- The network to which the server-assigned IP address belongs must be known in advance.

If any of these conditions are not met, the mode of assigning IP addresses to interfaces will have to be configured (this will be explained below).

As an interface address

The IP address assigned becomes the interface address configured using the *ADVANCED ADDRESS-AS-SIGNED-TO-IFC* command. This interface is normally a loopback interface.

Here you can see a typical sample configuration:

```
   log-command-errors
   no configuration
   set hostname bintec
   add device ppp 1
   add device loopback 1
   set data-link at cellular0/0
   set data-link at cellular0/1
   set data-link at cellular1/0
```

```
   set data-link at cellular1/1
   feature access-lists
; -- Access Lists user configuration --
      access-list 100
         entry 1 default
         entry 1 permit
         entry 1 source address interface loopback1
      exit
;
   exit
;
   global-profiles dial
; -- Dial Profiles Configuration --
      profile HSPA default
      profile HSPA inout
      profile HSPA 3gpp-apn movistar.es
;
      profile MOVISTAR default
      profile MOVISTAR dialout
      profile MOVISTAR idle-time 300
;
   exit
;
   network ethernet0/0
; -- Ethernet Interface User Configuration --
      ip address 192.6.2.186 255.255.255.0
;
   exit
;
   network cellular1/0
; -- Interface AT. Configuration  --
      pin plain 1111
      sim-select internal-socket-2
;
      network mode automatic
      network domain cs+ps
   exit
;
   network cellular1/1
; -- Interface AT. Configuration  --
      lcp-options acfc
      lcp-options pfc
      lcp-options accm 0
exit
;
   network ppp1
; -- Generic PPP User Configuration --
      ip address unnumbered
      ppp
; -- PPP Configuration --
         authentication sent-user MOVISTAR password keykey
         ipcp local address assigned
         no ipcp peer-route
         lcp echo-req off
      exit
;
      base-interface
; -- Base Interface Configuration --
         base-interface cellular1/1 link
         base-interface cellular1/1 profile HSPA
;
      exit
   exit
;
   network loopback1
; -- Loopback interface configuration --
      ip address unnumbered
```

```
   exit
;
   protocol ip
; -- Internet protocol user configuration --
      route 80.36.189.231 255.255.255.255 ppp1
      route 0.0.0.0 0.0.0.0 loopback1
;
      rule 2 local-ip loopback1 remote-ip any
      rule 2 napt translation
      rule 2 napt firewall
;
      rule 1 local-ip ppp1 remote-ip any
      rule 1 napt translation
      rule 1 napt firewall
;
      classless
;
      ipsec
; -- IPSec user configuration --
         enable
         assign-access-list 100
;
         template 1 isakmp tdes md5
         template 1 destination-address 80.36.189.231
         template 1 ike mode aggressive
         template 1 ike natt-version draft-v3
         template 1 config responder
         template 1 ike method xauth-init-preshared
         template 1 ike idtype keyed
         template 1 ike group two
;
         template 2 dynamic esp tdes md5
         template 2 source-address ppp1
         template 2 destination-address 80.36.189.231
;
         map-template 100 2
         key preshared ip 80.36.189.231 plain key1
         advanced address-assigned-to-ifc loopback1
;
         xauth-ip 80.36.189.231 user anonymous
         xauth-ip 80.36.189.231 password plain pppp
;
      exit
;
   exit
   dump-command-errors
   end
```

The series of events are as follows:

- The device starts up.
- As the device has a default route to the loopback interface, it performs NAPT when there is traffic (as described by rule 2) using the loopback1 interface as source address for the packets.
- The traffic then matches entry 1 of ACL 100 and negotiations begin with address 80.36.189.231. This address uses the ppp1 interface, setting the ppp1 IP address as source.
- The ISAKMP session is established and the assigned IP address becomes the loopback1 address.
- The SAs are established. All packets go through the SAs, given that traffic has loopback1 as route, applying rule 2, and setting the new loopback1 address as source and matching entry 1 of ACL 100.

Notes:

- All traffic receives IPsec protection with a server-assigned source address.
- You can access the device from the server using the assigned IP address.
- If you don't need a firewall in ppp1, rule 1 isn't necessary.

If you *only* want traffic with destination 172.24.0.0/16 to be protected by IPsec and everything else to go through ppp1, you should change the routes as follows:

```
route 172.24.0.0 255.255.0.0 loopback1
route 0.0.0.0 0.0.0.0 ppp1
```

That is, the traffic you want to protect must be specified in the routes. All traffic with a route pointing to loopback1 is protected by IPsec.

2.3.4 IKEv2 configuration mode

IKEv2 configuration mode allows peers to exchange route information during the IKE negotiation. This is achieved by sending routes via the configuration payload. IKEv2 configuration payloads are of type **cfg_request**, **cfg_reply**, **cfg_set** and **cfg_ack**.

Use the following parameters from the template menu to configure this method:

Command	Operation
CONFIG	Specifies whether the device initiates the configuration method.

“TEMPLATE [ID] CONFIG”

Enables the configuration mode. For the moment, IPv4 and IPv6 routes can be sent via the IKEv2 CFG payload. Configuring the *TEMPLATE <ID> CONFIG* command without any options enables the device to accept routes.

Example:

```
IPSec config>$template 1 config ?
  route-set       Enable sending routes via IKEv2 CFG payload
  route-accept    Enable accepting routes via IKEv2 CFG payload
  no              Negates a command or sets its default
  <cr>
IPSec config>template 1 config
IPSec config>
```

Command history:

Release	Modification
11.01.04	This command, with no options, was introduced as of version 11.01.04.

“TEMPLATE [ID] CONFIG ROUTE-ACCEPT”

Specifies whether the device accepts and installs the routes that it receives via the configuration mode. You can set the metric used for installing received routes.

Example:

```
IPSec config>template 1 config route-accept ?
  metric    Specify metric for routes
  <cr>
IPSec config>template 1 config route-accept metric ?
  <1..255>    Value in the specified range
IPSec config>template 1 config route-accept metric 30
IPSec config>
```

By default, the router accepts the routes that it receives. Use the **no** form of this command to stop the router from accepting and installing routes.

Example:

```
IPSec config>no template 1 config route-accept
  IPSec config>
```

Command history:

Release	Modification
11.01.04	This command was introduced as of version 11.01.04.

“TEMPLATE [ID] CONFIG ROUTE-SET”

Specifies the route-set parameters sent to the peer via the configuration mode. This command offers the following options:

Example:

```
IPSec config>template 1 config route-set ?
  interface           Send Interface route
  <1..99>             Send routes specified on an Access list (Standard)
  ipv6-access-list    Send routes specified on an IPv6 Access list (Standard)
IPSec config>template 1 config route-set 1
IPSec config>
```

“TEMPLATE [ID] CONFIG ROUTE-SET INTERFACE”

Specifies the route-set parameters sent to the peer via the configuration mode. An **internal_IP4_subnet** and/or **internal_IP6_subnet** attribute, containing the main IPv4 and/or IPv6 addresses of the interface protected by this template (assigned with the *TUNNEL-PROTECTION* command), are sent in the configuration payload.

Example:

```
IPSec config>template 1 config route-set interface
IPSec config>
```

Command history:

Release	Modification
11.01.04	This command was introduced as of version 11.01.04.

“TEMPLATE [ID] CONFIG ROUTE-SET <1..99>”

Specifies the route-set parameters sent to the peer via the configuration mode. An **internal_IP4_subnet** attribute containing the routes configured in the standard access-list is sent in the configuration payload.

Example:

```
IPSec config>template 1 config route-set 1
IPSec config>
```

Command history:

Release	Modification
11.00.07	This command was introduced as of version 11.00.07.
11.01.02	This command was introduced as of version 11.01.02.

“TEMPLATE [ID] CONFIG ROUTE-SET IPV6-ACCESS-LIST”

Specifies the route-set parameters sent to the peer via the configuration mode. An **internal_IP6_subnet** attribute, containing the routes configured in the IPv6 access-list, is sent in the configuration payload.

Example:

```
IPSec config>template 1 config route-set ipv6-access-list list1
IPSec config>
```

Command history:

Release	Modification
11.01.03	This command was introduced as of version 11.01.03.

2.3.5 GDOI group [id]

Configures a GDOI server where clients register to download policies and encryption keys. The *GDOI GROUP <id>* command gives you access to the GDOI server configuration menu where you will find the following commands:

Command	Function
ADDRESS IPV4 <ip>	Sets the server local IP address.
IDENTITY NUMBER <id>	Sets the group identifier.
REKEY ADDRESS IPV4 <ip>	Sets the multicast IP for rekey messages.
REKEY ALGORITHM <alg>	Sets the encryption algorithm to use in rekey messages (des, 3des or aes).

REKEY AUTHENTICATION RSA	Sets the RSA key to use for authentication in rekey messages.
REKEY LIFETIME SECONDS <sec>	Sets the rekey SA lifetime.
REKEY RETRANSMIT <s> <n>	Sets the rekey message retransmission interval and the number of retransmissions.
REKEY TRANSPORT UNICAST	Rekey messages are sent to the client IP.
REKEY TRANSPORT MULTICAST	Rekey messages are sent to a multicast IP.
SA IPSEC <id>	Allows you to enter a SA configuration menu.

You can use the *SA IPSEC <sa-id>* command to access the configuration submenu of a specific SA.

Example:

```
IPSec GDOI config>sa ipsec 1
GDOI SA config>
```

Said submenu contains the following commands:

Command	Function
ENCAPSULATION TUNNEL	Sets tunnel mode as the encapsulation type for clients. This is the default option.
ENCAPSULATION TRANSPORT	Sets transport mode as the encapsulation type for clients.
LIFETIME <time>	Sets the lifetime on SAs created by clients.
MATCH ADDRESS IPV4 <acclst>	Configures the access list used by clients registered on this server.
REPLAY COUNTER	Enables the sequence number anti-replay mechanism.
REPLAY NONE	Disables anti-replay.
REPLAY TIME	Enables timestamp anti-replay. This is the default option.
TRANSFORM-SET <ciph> <auth>	Defines the encryption and authentication algorithm to be used by the SAs created by clients.

Example:

GDOI server group 2. &&&&&

The rekey is configured in multicast mode using address 239.0.0.2, while 256-bit AES is used as the encryption algorithm for the rekey packets. RSA MYKEY (which you must generate beforehand using the *KEY RSA GENERATE* command) is used to authenticate the rekey messages. The AES keys used for the rekey are renewed every ten minutes, with three retransmissions sent every 10 seconds.

As for the SAs, a SA has been configured and associated with a preconfigured access list (100) (please see manual *bintec Dm752-I Access Control*). The traffic belonging to this SA is encrypted an TripleDES and authenticated using SHA. The keys are valid for 5 minutes, after which the server sends a rekey message to renew them. The IPsec tunnel mode is used for encapsulation.

```
         gdoi group 2
; -- GDOI user configuration --
            identity number 2
            rekey address ipv4 239.0.0.2
            rekey algorithm aes-256
            rekey authentication rsa MYKEY
            rekey lifetime seconds 10m
            rekey retransmit 10s number 3
            sa ipsec 1
               lifetime 5m
               match address ipv4 100
               transform-set tdes sha1
            exit
;
exit
```

2.3.6 Fault-tolerant

The *FAULT-TOLERANT* command provides access to the IPsecFT protocol configuration submenu.

Depending on the operating mode of the router, the commands in this submenu will vary. That is, the commands available to you when the router is acting as master will be different to the ones you can use when the router is acting as slave. The commands for both modes of operation are shown below:

Slave function commands

Commands for slave	Function
ENABLE	Enables IPsecFT.
LIST	Lists IPsecFT configuration.
LISTEN-PORT	Listening port for incoming IPsecFT connections.
MODE	Changes the operating mode between master and slave.
NO	Restores a command to its default value.
EXIT	Exits IPsecFT configuration menu.

Master function commands

Commands for master	Function
ENABLE	Enables IPsecFT.
INHERIT-CONDITION	Selects the condition for sending IPsecFT database sessions to IPsec.
LIST	Lists IPsecFT configuration.
LISTEN-PORT	Listening port for incoming IPsecFT connections.
MODE	Changes the operating mode between master and slave.
NO	Restores a command to its default value.
SLAVE-ADDRESS	IP address for connecting to the slave.
SLAVE-PORT	Port for connecting to the slave.
SOURCE-ADDRESS	Source address to use when sending IPsecFT packets.
TIMERS	Sets IPsecFT wait times.
EXIT	Exits IPsecFT configuration menu.

“ENABLE”

Enables IPsecFT.

The *SOURCE-ADDRESS* and *SLAVE-ADDRESS* commands must be configured before IPsecFT is enabled or you will receive an error message.

IPsecFT is disabled whenever you use the *MODE* command to change the operating mode.

Example:

```
IPSecFT config>enable
CLI Error: Source address is not configured. Unable to enable
CLI Error: Command error
```

The example attempts to enable IPsecFT without configuring the IPsecFT packet source.

Example:

```
IPSecFT config>enable
CLI Error: Slave address is not configured. Unable to enable
CLI Error: Command error
```

The example attempts to enable IPsecFT without configuring the slave IP address.

Example:

```
IPSecFT config>enable
```

This example enables IPsecFT.

“INHERIT-CONDITION VRRP”

Specifies the condition that is used to tell IPsec to set up the sessions contained in the IPsecFT database. Please remember that these sessions are the result of master and slave exchanging data packets in IPsecFT.

Example:

```
IPSecFT config>inherit-condition vrrp
```

This example selects the VRRP that decides when to tell IPsec to set up IPsecFT database sessions.

“LIST”

Lists the protocol configuration. The amount of information shown will vary depending on the operating mode.

Example:

```
IPSecFT config>list
Fault tolerant configuration:
        Enable: TRUE
        Mode: Master
        Slave server address: 1.1.1.1
        Slave server port: 52912
        Source address: ethernet0/1
        Listen port: 52912
        Inactivity timeout: 500 milliseconds
        Keepalive period: 100 milliseconds
        Inherit condition: VRRP
```

This example lists the configuration in master mode.

Example:

```
IPSecFT config>list
Fault tolerant configuration:
         Enable: FALSE
         Mode: Slave
         Listen port: 52912
```

This example lists the configuration in slave mode.

“LISTEN-PORT [PORT]”

Listening port for incoming IPsecFT connections. When operating in master mode, the value of this command is sent to the slave device so that it initiates the connection with the port.

The default port value is 52912.

Example:

```
IPSecFT config>listen-port 5645
```

The example sets the listening port to 5645.

“MODE {MASTER/SLAVE}”

Selects the IPsecFT protocol operating mode, which can be either master or slave.

All the IPsecFT parameters are configured in the master device. These are sent to the slave device when it connects to the master.

The slave device is only configured with the port that listens for the IPsecFT connections. When the first connection is accepted, the slave device receives the parameters to initiate a new IPsecFT connection in the opposite direction. The slave uses the parameters it received in the last connection when attempting to establish IPsecFT sessions.

Example:

```
IPSecFT config>mode master
```

This example configures the device as master.

“SLAVE-ADDRESS [IP-ADDRESS]”

Configures the IP address that connects to the slave.

Example:

```
IPSecFT config>slave-address 1.1.1.1
```

This example configures the slave IP address as 1.1.1.1.

“SLAVE-PORT [PORT]”

Configures the port to connect to the slave.

The default (port) is 52912.

Example:

```
IPSecFT config>slave-port 4658
```

This example configures the slave port as 4658.

“SOURCE-ADDRESS [IP-ADDRESS/INTERFACE]”

Configures the master with either the source address to use in IPsecFT packets or the interface through which IPsecFT packets are to be sent. The source address used in the slave will always be the destination address of the first packet in the IPsecFT session establishment.

The two devices that make up the fault tolerant IPsec recovery system use the IPsecFT session to exchange data. This IPsecFT session must be established in a controlled manner, along a path that the user deems appropriate. You can use the *SOURCE-ADDRESS* command to ensure the session is established using the selected source.

Example:

```
IPSecFT config>source-address ethernet0/0
```

This example configures the IPsecFT packets to be sent through the ethernet0/0 interface.

“TIMERS KEEPALIVE-PERIOD [KEEPALIVE] INACTIVITY-TIMEOUT [INACTIVITY]”

Configures the amount of time the IPsecFT protocol waits before taking decisions.

The *keepalive* value refers to the amount of time the protocol waits before it's next action, that is, how long it waits before sending monitoring packets, polling the VRRP, or changing states.

The inactivity value configures the maximum time to wait for a packet from the other end before considering the IPsecFT session down.

Low values in this command can lead to high CPU usage.

The *Keepalive* default is 100 milliseconds and the *inactivity* default is 500 milliseconds.

Example:

```
timers keepalive-period 200 inactivity-timeout 1000
```

This example configures a keepalive value of 200 milliseconds and an inactivity value of 1 second.

2.3.7 Report IPsec statistics

IPsec statistics can be reported to NSLA filters to dynamically change the behavior of other functionalities. For more information on the NSLA feature, read the *bintec Dm754-I NSLA* manual.

The *REPORT* command is provided for this purpose. The following options can be used:

```
IPSec config>report ?
  rx-rate-kbps    Report the input rate (Kbps)
  tx-rate-kbps    Report the output rate (Kbps)
```

“REPORT RX-RATE-KBPS NSLA-FILTER [1..65535] INTERVAL-SEC [1..3600]”

Periodically sends a report on the received rate in kilobits per second to an NSLA filter.

Example:

```
IPSec config>report rx-rate-kbps nsla-filter 3 interval-sec 5
```

Use *NO REPORT RX-RATE-KBPS* to disable the periodic report.

Command history:

Release	Modification
11.00.05	This command was introduced as of version 11.00.05.
11.01.01	This command was introduced as of version 11.01.01.

“REPORT TX-RATE-KBPS NSLA-FILTER [1..65535] INTERVAL-SEC [1..3600]”

Periodically sends a report on the transmitted rate in kilobits per second to an NSLA filter.

Example:

```
IPSec config>report tx-rate-kbps nsla-filter 3 interval-sec 5
```

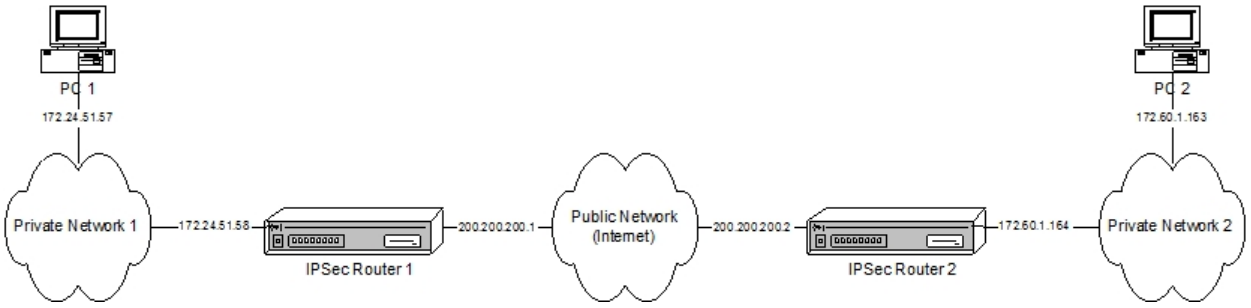
Use *NO REPORT TX-RATE-KBPS* to disable the periodic report.

Command history:

Release	Modification
11.00.05	This command was introduced as of version 11.00.05.
11.01.01	This command was introduced as of version 11.01.01.

2.4 Examples

2.4.1 Example 1: Manual mode



The idea is to create a new virtual private network (VPN) between host A and host B. All other private network traffic passes normally. The traffic from A to B will be encrypted through an IPsec tunnel using the triple DES encryption algorithm and SHA1 authentication.

2.4.1.1 Creating access control lists

As already mentioned, the tunnel clients are host A and host B.

Router 1:

```
Config>feature access-lists

-- Access Lists user configuration --
Access Lists config>access-list ?
  <1..99>       Standard Access List number (1-99)
  <100..199>    Extended Access List number (100-1999)
  <5000..9999>    Stateful access-list
Access Lists config>access-list 101

Extended Access List 101>entry 1 source address 172.24.51.57 255.255.255.255
Extended Access List 101>entry 1 destination address 172.60.1.163 255.255.255.255
Extended Access List 101>
```

The access list is configured as follows:

```
Extended Access List 101>list all-entries
Extended Access List 101, assigned to no protocol
1     PERMIT  SRC=172.24.51.57/32  DES=172.60.1.163/32  Conn:0
Extended Access List 101>
```

You can use the *SHOW CONFIG* command to display the configuration and use it later. All you have to do is enter the command in the console as shown below:

```
Access Lists config>show config
; Showing Menu and Submenus Configuration for access-level 15 ...

      access-list 101
         entry 1 default
         entry 1 permit
         entry 1 source address 172.24.51.57 255.255.255.255
```

```
         entry 1 destination address 172.60.1.163 255.255.255.255
;
      exit
;
Access Lists config>
```

Please note that the source and destination addresses used for the Router 1 example are the other way round in the Router 2 example.

Router 2:

```
Access Lists config>show config
; Showing Menu and Submenus Configuration for access-level 15 ...

      access-list 101
         entry 1 default
         entry 1 permit
         entry 1 source address 172.60.1.163 255.255.255.255
         entry 1 destination address 172.24.51.57 255.255.255.255
;
      exit
;
Access Lists config>
```

2.4.1.2 Creating templates

Here we create the security patterns or templates:

Router 1:

First we have to enable IPsec:

```
Config>protocol ip
-- Internet protocol user configuration --
IP config>ipsec

-- IPSec user configuration --
IPSec config>enable
IPSec config>
```

Now we configure the template:

```
IPSec config>template 2 ?
  manual     Manual template
  isakmp     Isakmp template
  dynamic    Dynamic template
IPSec config
IPSec config>template 2 manual ?
  esp      ESP security service (Encapsulating Security Payload)
  ah       AH security service (Authentication Header)
IPSec config>template 2 manual esp ?
  des       encryption algorithm DES (Data Encryption Standard)
  tdes      encryption algorithm TDES (Triple Data Encryption Standard)
IPSec config>template 2 manual esp tdes ?
  md5       authentication algorithm MD5
  sha1      authentication algorithm SHA1
  sha256    authentication algorithm SHA256
  sha384    authentication algorithm SHA384
  sha512    authentication algorithm SHA512
  none      no authentication algorithm
IPSec config>template 2  ?
  manual                 Manual template
  source-address         Tunnel's local IP address
  destination-address    Address of the other remote end of the tunnel
  spi                    Security Parameter Index
  key                    Template encryption DES key
  tkey                   Triple DES key
  md5key                 MD5 key
  sha1key                SHA1 key
  antireplay             Activates the Anti-Replay service
```

```
  df-bit                 Treatment of the Don't Fragment bit
  mtu-threshold          Set minimum MTU in bytes (default 576)
  mtu-default            Set starting value for path MTU
  no                     Negates a command or sets its default
IPSec config>template 2 source-address ?
  <a.b.c.d>      Ipv4 format
  <interface>    Interface name
IPSec config>template 2 source-address 200.200.200.1
IPSec config>template 2 destination-address ?
  <a.b.c.d>      Ipv4 format
  <word>         Text
IPSec config>template 2 destination-address 200.200.200.2
IPSec config>template 2 spi ?
  <257..65535>    Enter SPI (SPI > 256):
IPSec config>template 2 spi 280
IPSec config>template 2 tkey h53s45ef46agv4646n2j8qpo

IPSec config>template 2 sha1key b74hd748ghzm67k6m6d1
```

The template configuration looks like this:

```
IPSec config>list template all
TEMPLATES
2 manual ESP-3DES ESP-SHA1  SRC=200.200.200.1 DES=200.200.200.2 SPI=280

IPSec config>
```

And the *SHOW CONFIG* command will give us:

```
IPSec config>show config
; Showing Menu and Submenus Configuration for access-level 15 ...

         enable
;
         template 2 manual esp tdes sha1
         template 2 source-address 200.200.200.1
         template 2 destination-address 200.200.200.2
         template 2 spi 280
         template 2 tkey 0x6835337334356566343661677634363436E326A3871706F
         template 2 sha1key 0x623734686437343867687A6D36376B366D366431
;
IPSec config>
```

Router 2:

```
IPSec config>show config
; Showing Menu and Submenus Configuration for access-level 15 ...

         enable
;
         template 2 manual esp tdes sha1
         template 2 source-address 200.200.200.2
         template 2 destination-address 200.200.200.1
         template 2 spi 280
         template 2 tkey 0x6835337334356566343661677634363436E326A3871706F
         template 2 sha1key 0x623734686437343867687A6D36376B366D366431
;
IPSec config>
```

Please note that the source and destination addresses used for the Router 1 example are the other way round in the Router 2 example.

Note

Both routers must be configured with the same SPI value.

2.4.1.3 Creating the SPDs

To complete the security policy databases (SPD), we now need to map the access control list (ACL) elements to the chosen templates.

Router 1:

```
IPSec config>assign-access-list ?
  <100..1999>    Enter extended access list id
IPSec config>assign-access-list 101
IPSec config>map-template ?
  <100..1999>    Enter extended access list id
IPSec config>map-template 101 ?
  <1..65535>    Enter template id(1-65534)
IPSec config>map-template 101 2
IPSec config>
```

The IPsec configuration looks like this:

```
IPSec config>list all


IPSec Access Control.
Access Control is: enabled
QOS Preclassify is: disabled

Access Lists assigned to IPSec:
  Extended Access List 101
  Templates: 2


Extended Access List 101, assigned to IPSec

1     PERMIT  SRC=172.24.51.57/32  DES=172.60.1.163/32  Conn:0

TEMPLATES
2 manual ESP-3DES ESP-SHA1  SRC=200.200.200.1 DES=200.200.200.2 SPI=280

0 key entries
0 rsakey entries
Id.           Date.         Len         Certificate Authority (CA)           Cert sn.

Period to check LDAP servers not configured. Using default value: 24h0m0s

KeepAlive Configuration:
   Maximum number of encoded packets without receiving an answer: 0.
   Timeout after last packet encoded: 0 seconds.

DPD Configuration:
Idle period(secs) before sending DPD keepalives: 60
Maximum number of DPD keepalives not acknowledged: 3
Period of time(secs) between DPD keepalives: 5
Always send keepalive after idle period expiration : DISABLED
Anti-replay : DISABLED

Hash Configuration:
Maximum number of entries in hash table: 50000
Shift constant for exponential moving average calculation: 8

Check-out time (%) - from SA's end-lifetime - to renegotiate : 10

SA's purge timeout:  15

Period of connected SA's notification event:  0

NAT Translation Port:  4500

Use software exponentiation
```

```
Maximum number of IPSec headers in a packet:  1

IPSec config>
```

The *SHOW CONFIG* command gives us the following:

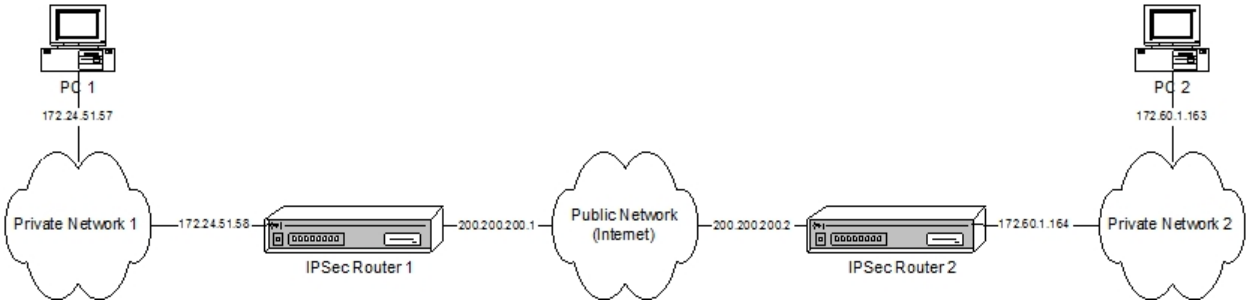
```
IPSec config>show config
; Showing Menu and Submenus Configuration for access-level 15 ...

        enable
        assign-access-list 101
;
        template 2 manual esp tdes sha1
        template 2 source-address 200.200.200.1
        template 2 destination-address 200.200.200.2
        template 2 spi 280
        template 2 tkey 0x68353373343565663436616776343634366E326A3871706F
        template 2 sha1key 0x623734686437343867687A6D36376B366D366431
;
        map-template 101 2
IPSec config>
```

Router 2:

```
IPSec config>show config
; Showing Menu and Submenus Configuration for access-level 15 ...

        enable
        assign-access-list 101
;
        template 2 manual esp tdes sha1
        template 2 source-address 200.200.200.2
        template 2 destination-address 200.200.200.1
        template 2 spi 280
        template 2 tkey 0x68353373343565663436616776343634366E326A3871706F
        template 2 sha1key 0x623734686437343867687A6D36376B366D366431
;
        map-template 101 2
IPSec config>
```

Now hosts A and B can communicate securely as far as the communication is concerned. However, it is up to the user to ensure the complete security of the communications system, which is also based on the devices, keys, modification permissions, etc.

2.4.2 Example 2: Dynamic mode (IPSEC IKE main mode)



The scenario here is the same as the previous one, except that the tunnel will be based on dynamic templates. Consequently, any communications, keys, etc., will be automatically negotiated in main mode.

2.4.2.1 Creating the access control lists

The configuration is the same as for example 1.

2.4.2.2 Creating templates

Now we need to create the ISAKMP and dynamic templates. Please note that the pre-shared key in the final command must be the same in both devices. Main mode, which masks the identities of the tunnel end routers, is used as the default negotiation mode. While the same lifetimes have been entered in both devices, they can be changed and negotiated.

Router 1:

```
IPSec config>enable
IPSec config>template 1 ?
  manual     Manual template
  isakmp     Isakmp template
  dynamic    Dynamic template
IPSec config>template 1 isakmp ?
  des       Encryption algorithm DES (Data Encryption Standard)
  tdes      Encryption algorithm TDES (Triple Data Encryption Standard)
  aes128    Encryption algorithm AES using 128-bit key (Advanced Encryption
            Standard)
  aes192    Encryption algorithm AES using 192-bit key (Advanced Encryption
            Standard)
  aes256    Encryption algorithm AES using 256-bit key (Advanced Encryption
            Standard)
IPSec config>template 1 isakmp tdes ?
  md5       Authentication algorithm MD5
  sha1      Authentication algorithm SHA1
  sha256    Authentication algorithm SHA256
  sha384    Authentication algorithm SHA384
  sha512    Authentication algorithm SHA512
IPSec config>template 1 isakmp tdes sha1
IPSec config>template 1  ?
  isakmp                 Isakmp template
  destination-address    Address of the other remote end of the tunnel
  discover               Use TED to discover the remote end of the tunnel
  backup-destination     Backup destination IP address
  disable-fast-return-from-backup    Disable checking of Main availability and
                                     fast return to Main from Backup
  udp-encapsulation      Enables UDP encapsulation
  udp-ike                Enables IKE UDP encapsulation
  life                   Introduces the SAs life span created from the
                         template
  ike                    Configures parameters relative to the IPSec IKE mode
  keepalive              Enables the available keepalive services
  config                 Isakmp configuration
  aggressive             Aggressive configuration mode ciphered/clear
  send-original-pkt      Send original packet after tunnel establishment
  set-label              Set Label Value of IKE and IPSEC packets
  no                     Negates a command or sets its default
IPSec config>template 1 destination-address ?
  <a.b.c.d>      Ipv4 format
  <word>         Text
IPSec config>template 1 destination-address 200.200.200.2
IPSec config>template 1 life ?
  type        Type of life duration for the SA
  duration    Life duration
IPSec config>template 1 life duration ?
  seconds    Lifetime in seconds
  kbytes     Lifetime in kbytes
IPSec config>template 1 life duration seconds ?
  <2m..3550w>    Time value
IPSec config>template 1 life duration seconds 12h
IPSec config>template 3 ?
IPSec config>template 3 dynamic ?
  esp               ESP security service (Encapsulating Security Payload)
  ah                AH security service (Authentication Header)
  gdoi-ks-policy    Encryption and authentication algorithms received from
                    GDOI Key Server
```

```
IPSec config>template 3 dynamic esp ?
  des       Encryption algorithm DES (Data Encryption Standard)
  tdes      Encryption algorithm TDES (Triple Data Encryption Standard)
  aes128    Encryption algorithm AES using 128-bit key (Advanced Encryption
            Standard)
  aes192    Encryption algorithm AES using 192-bit key (Advanced Encryption
            Standard)
  aes256    Encryption algorithm AES using 256-bit key (Advanced Encryption
            Standard)
IPSec config>template 3 dynamic esp tdes ?
  md5       Authentication algorithm MD5
  sha1      Authentication algorithm SHA1
  sha256    Authentication algorithm SHA256
  sha384    Authentication algorithm SHA384
  sha512    Authentication algorithm SHA512
  none      No authentication algorithm
IPSec config>template 3 dynamic esp tdes md5
IPSec config>template 3 ?
  dynamic                         Dynamic template
  source-address                  Tunnel's local IP address
  destination-address             Address of the other remote end of the
                                  tunnel
  antireplay                      Activates the Anti-Replay service
  padding-check                   Enables padding check
  life                            Introduces the SAs life span created from
                                  the template
  ike                             Configures parameters relative to the IPSec
                                  IKE mode
  keepalive                       Enables the available keepalive services
  encap                           Type of encapsulation for packets
  napt-id-skipped                 Ipsec must not mark packets for napt
  fast-forwarder                  Force fast-forwarding of packets
  invalid-spi-recovery            Enables invalid SPI recovery
  df-bit                          Treatment of the Don't Fragment bit
  mtu-threshold                   Set minimum MTU in bytes (default 576)
  mtu-default                     Set starting value for path MTU
  tcp-mss-adjust                  Adjust the mss of transit packets
  rri-enabled                     Enables RRI (Reverse Route Injection)
  rri-nexthop                     Chooses RRI's next-hop to use
  mapped-to-ifc                   Maps the template to an ifc
  direct-encoded-fwd              Send encoded packets directly to an ifc
  assigned-address-goes-to-ifc    The received address during isakmp config
                                  will be assigned to this ifc
  mapped-to-ifc                   Maps the template to an ifc
  assigned-address-goes-to-ifc    The received address during isakmp config
                                  will be assigned to this ifc
  unique                          Only one similar tunnel allowed per
                                  access-list entry
  pkt-src-client-src              Use original packet source as client source
  replace-destination             Replace packets destination with tunnel
                                  destination
  prefragmentation                Enables the prefragmentation process
  gdoi                            Group Domain of Interpretation configuration
  set-label                       Set Label Value of IKE and IPSEC packets
  vrf                             Assigns the template to a VRF
  fault-tolerant                  Assigns the template to the fault tolerant
                                  IPSec recovery system
  report                          Report statistics
  no                              Negates a command or sets its default
IPSec config>template 3 source-address ?
  <a.b.c.d>      Ipv4 format
  <interface>    Interface name
IPSec config>template 3 source-address 200.200.200.1
IPSec config>template 3 destination-address ?
  <a.b.c.d>      Ipv4 format
  <word>         Text
IPSec config>template 3 destination-address 200.200.200.2
```

```
IPSec config>emplate 3 life ?
  type        Type of life duration for the SA
  duration    Life duration
IPSec config>template 3 life type ?
  seconds    Lifetime in seconds
  kbytes     Lifetime in kbytes
  both       Lifetime in seconds and kbytes
IPSec config>template 3 life type both
IPSec config>template 3 life duration ?
  seconds    Lifetime in seconds
  kbytes     Lifetime in kbytes
IPSec config>template 3 life duration seconds ?
  <2m..3550w>    Time value
IPSec config>template 3 life duration seconds 4h
IPSec config>template 3 life duration kbytes ?
  <0..4294967295>    kbytes
IPSec config>template 3 life duration kbytes 0
IPSec config>key preshared ip 200.200.200.2 plain 1234567890123456
IPSec config>
```

We could also have made use of the text-mode configuration (starting with a *SHOW CONFIG* output).

```
IPSec config>show config
; Showing Menu and Submenus Configuration for access-level 15 ...

        enable
;
        template 1 isakmp tdes sha1
        template 1 destination-address 200.200.200.2
        template 1 life duration seconds 12h
        template 1 ike natt-version rfc
        template 1 keepalive dpd
;
        template 3 dynamic esp tdes md5
        template 3 source-address 200.200.200.1
        template 3 destination-address 200.200.200.2
        template 3 life type both
        template 3 life duration seconds 4h
        template 3 life duration kbytes 0
;
        key preshared ip 200.200.200.2 ciphered 0xE21C47018BC8B868FB72F48DC4363FC0
CFABF60C9FFE0286
        advanced dpd no always-send
IPSec config>
```

Router 2:

```
IPSec config>show config
; Showing Menu and Submenus Configuration for access-level 15 ...

        enable
;
        template 1 isakmp tdes sha1
        template 1 destination-address 200.200.200.1
        template 1 life duration seconds 12h
        template 1 ike natt-version rfc
        template 1 keepalive dpd
;
        template 3 dynamic esp tdes md5
        template 3 source-address 200.200.200.2
        template 3 destination-address 200.200.200.1
        template 3 life type both
        template 3 life duration seconds 4h
        template 3 life duration kbytes 0
;
        key preshared ip 200.200.200.1 ciphered 0xE21C47018BC8B868FB72F48DC4363FC0
CFABF60C9FFE0286
        advanced dpd no always-send
```

```
IPSec config>
```

2.4.2.3 Creating the SPDs

Finally, we need to establish the SPDs:

Router 1:

```
IPSec config>assign-access-list ?
  <100..1999>    Enter extended access list id
IPSec config>assign-access-list 101
IPSec config>map-template ?
  <100..1999>    Enter extended access list id
IPSec config>map-template 101 ?
  <1..65535>    Enter template id(1-65534)
IPSec config>map-template 101 3
IPSec config>
```

The final IPsec configuration looks like this:

```
IPSec config>list all


IPSec Access Control.
Access Control is: enabled
QOS Preclassify is: disabled

Access Lists assigned to IPSec:
  Extended Access List 101
  Templates: 3


Extended Access List 101, assigned to IPSec

1     PERMIT  SRC=172.24.51.57/32  DES=172.60.1.163/32  Conn:0

TEMPLATES
1 isakmp 3DES SHA1  SRC=internal DES=200.200.200.2
   LifeTime:12h0m0s
   IKE MAIN
   PRESHARED
   addr4 ID TYPE
   OAKLEY GROUP 1
   DPD enabled

3 dynamic ESP-3DES ESP-MD5  SRC=200.200.200.1  DES=200.200.200.2
   LifeTime:4h0m0s 0 kbytes
   PFS disabled

1 key entries
   200.200.200.2 ************
0 rsakey entries
Id.           Date.         Len         Certificate Authority (CA)           Cert sn.

Period to check LDAP servers not configured. Using default value: 24h0m0s

KeepAlive Configuration:
   Maximum number of encoded packets without receiving an answer: 0.
   Timeout after last packet encoded: 0 seconds.

DPD Configuration:
Idle period(secs) before sending DPD keepalives: 60
Maximum number of DPD keepalives not acknowledged: 3
Period of time(secs) between DPD keepalives: 5
Always send keepalive after idle period expiration : DISABLED
Anti-replay : DISABLED

Hash Configuration:
```

```
Maximum number of entries in hash table: 50000
Shift constant for exponential moving average calculation: 8

Check-out time (%) - from SA's end-lifetime - to renegotiate : 10

SA's purge timeout:  15

Period of connected SA's notification event:  0

NAT Translation Port:  4500

Use software exponentiation

Maximum number of IPSec headers in a packet:  1

IPSec config>
```

The *SHOW CONFIG* command gives us:

```
IPSec config>show config
; Showing Menu and Submenus Configuration for access-level 15 ...

         enable
         assign-access-list 101
;
         template 1 isakmp tdes sha1
         template 1 destination-address 200.200.200.2
         template 1 life duration seconds 12h
         template 1 ike natt-version rfc
         template 1 keepalive dpd
;
         template 3 dynamic esp tdes md5
         template 3 source-address 200.200.200.1
         template 3 destination-address 200.200.200.2
         template 3 life type both
         template 3 life duration seconds 4h
         template 3 life duration kbytes 0
;
         key preshared ip 200.200.200.2 ciphered 0xE21C47018BC8B868FB72F48DC4363FC0
CFABF60C9FFE0286
         advanced dpd no always-send
IPSec config>
```

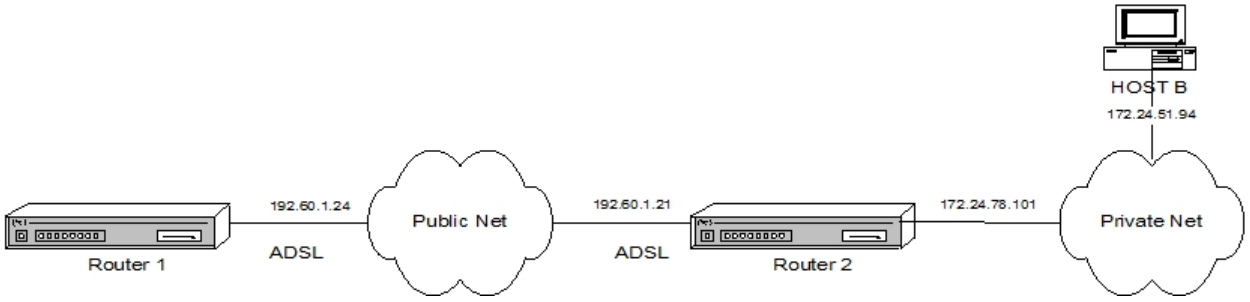
Router 2:

```
IPSec config>assign-access-list 101
IPSec config>map-template 101 3
IPSec config>show config
; Showing Menu and Submenus Configuration for access-level 15 ...

         enable
         assign-access-list 101
;
         template 1 isakmp tdes sha1
         template 1 destination-address 200.200.200.1
         template 1 life duration seconds 12h
         template 1 ike natt-version rfc
         template 1 keepalive dpd
;
         template 3 dynamic esp tdes md5
         template 3 source-address 200.200.200.2
         template 3 destination-address 200.200.200.1
         template 3 life type both
         template 3 life duration seconds 4h
         template 3 life duration kbytes 0
;
         key preshared ip 200.200.200.1 ciphered 0xE21C47018BC8B868FB72F48DC4363FC0
CFABF60C9FFE0286
```

```
        advanced dpd no always-send
IPSec config>
```

Hosts A and B can now communicate securely. The pre-shared key is the only key to protect in this case.

2.4.3 Example 3: Dynamic mode (IPSEC IKE aggressive mode) where the address of one of the tunnel endpoints is unknown



This scenario shows how to connect two routers via a virtual private network (VPN) using an ADSL line as the means of connection. Traffic will be encrypted through an IPsec tunnel based on dynamic templates with DES encryption and MD5 authentication used in the ISAKMP negotiation, and ESP with DES encryption and SHA1 authentication in the SA negotiation. By basing the tunnel on dynamic templates, any communications, keys, etc., will be automatically negotiated using aggressive mode.

The advantage of aggressive mode is that Router 2 doesn't need to know the IP address of the other end of the tunnel: This means that this is a suitable configuration for multiple devices to connect to one router (Router 2) simply knowing the *hostname* and the shared key. Router 1, on the other hand, does need to know the address of the router with which it is going to set up a tunnel. This is because it is the one that initiates the negotiation and therefore has to know which IP address to connect to.

In the following sections, we will first describe how to configure Router 1 before going on to configure Router 2 and explain any parameters that differ from the Router 1 configuration.

2.4.3.1 Configuring Router 1

2.4.3.1.1 Configuring the host name, IP addresses and rules

As noted above, the router uses the *hostname* (rather than the IP address) to authenticate. Therefore, the first thing we have to configure is the device name.

```
bintec               (c)1996-2002
*PROCESS 4
Config>set hostname GAS1
```

Next, we need to assign the IP address to the ADSL interface. We should also add a static route, specifying that all traffic to the private network is to be sent using the other end of the IPsec tunnel as the gateway.

We also have the option to specify a connection identifier for traffic between routers, though this is only necessary when you want packets in different connections to be treated differently.

```
GAS1 Config>list devices
 Interface       Con    Type of interface             CSR     CSR2  int
 ethernet0/0     LAN1   Quicc Ethernet           fa200a00 fa203c00   5e
 serial0/0       WAN1   X25                      fa200a20 fa203d00   5d
 atm0/0          ADSL1  Async Transfer Mode      fa200a60 fa203f00   55
 bri0/0          ISDN1  ISDN Basic Rate Int      fa200a40 fa203e00   5c
 x25-node        ---    Router->Node                    0             0
 ppp1            ---    Generic PPP                     0             0
 ppp2            ---    Generic PPP                     0             0
GAS1 Config>
```

2.4.3.1.2 Creating access control lists

Having configured all the IP parameters, we can now turn to the IPsec configuration itself.

First of all, we have to configure the access control lists (ACL). We do this by accessing the generic lists setup menu, picking a number from the list that corresponds to an extended list (between 100 and 199), indicating an entry ID from the list (in this case 1), and entering the desired values for the following parameters:

- The source IP address, i.e., the one already configured on the ADSL interface.

- The destination IP, i.e., the device with which we are going to set up an IPsec tunnel, which in our case is Router 2.
- The connection: we must indicate the connection ID assigned to tunnel traffic. You can display this ID using the *LIST RULE* command. In this particular example, however, we don't need to assign the connection ID because the routers don't apply treatment to packets based on the connection.
- The action to be taken on the packets, in our case, IPsec processing (**PERMIT**).

```
GAS1 Config>feature access-lists
-- Access Lists user configuration --
GAS1 Access Lists config>access-list ?
  <1..99>       Standard Access List number (1-99)
  <100..1999>   Extended Access List number (100-1999)
  <5000..9999>  Stateful access-list
GAS1 Access Lists config>access-list 102

GAS1 Extended Access List 102>entry 1 ?
  default           Sets default values to an existing or a new entry
  permit            Configures type of entry or access control as permit
  deny              Configures type of entry or access control as deny
  source            Source menu: subnet or port
  destination       Destination menu: subnet or port
  protocol          Protocol
  protocol-range    Protocol range
  connection        IP connection identifier (rule)
  description       Sets a description for the current entry
  ds-field          DSCP in IP packets
  label             Label for classification
  precedence        Precedence in IP packets
  tcp-specific      Tcp specific filtering
  tos-octet         TOS octet value in IP packets
  no                Negate a command or set its defaults
GAS1 Extended Access List 102>entry 1 source ?
  address       IP address and mask of the source subnet
  port-range    Source port range
GAS1 Extended Access List 102>entry 1 source address ?
  <a.b.c.d>      Ipv4 format
  <interface>    Interface name
GAS1 Extended Access List 102>entry 1 source address 192.60.1.24 ?
  <a.b.c.d>    Ipv4 format
GAS1 Extended Access List 102>entry 1 source address 192.60.1.24 255.255.255.255
GAS1 Extended Access List 102>entry 1 destination ?
  address       IP address and mask of the destination subnet
  port-range    Destination port range
GAS1 Extended Access List 102>entry 1 destination address ?
  <a.b.c.d>      Ipv4 format
  <interface>    Interface name
GAS1 Extended Access List 102>entry 1 destination address 172.24.0.0 ?
  <a.b.c.d>    Ipv4 format
GAS1 Extended Access List 102>entry 1 destination address 172.24.0.0 255.255.0.0
GAS1 Extended Access List 102>entry 1 permit
GAS1 Extended Access List 102>
```

The result of the configured access lists:

```
GAS1 Access Lists config>show config
; Showing Menu and Submenus Configuration for access-level 15 ...

      access-list 102
         entry 1 default
         entry 1 permit
         entry 1 source address 192.60.1.24 255.255.255.255
         entry 1 destination address 172.24.0.0 255.255.0.0
;
      exit
;
GAS1 Access Lists config>
```

2.4.3.1.3 Creating templates

Now we need to create the ISAKMP and dynamic templates. Please note that the pre-shared key in the final command must be the same in both devices. Also note that this example differs from the previous one in that negotiation is carried out in aggressive mode. This means the identities of the tunnel endpoint routers are not masked and the IP address of the other end of the tunnel is unknown.

Although we have also entered the same lifetimes in both tunnel endpoints, these parameters can be negotiated so that the the lowest value configured is used.

When creating the ISAKMP template, we have to specify the encryption (DES) and authentication (MD5) type to use (as indicated in the initial security specifications).

When creating the template, we must specify an ID number. The template ID number will be used throughout the template configuration. We also have to indicate the tunnel destination IP to which we are connecting and the mode that will be used. In this case the mode is aggressive mode, since the host name, rather than the IP address, will be used for authentication. This is very useful when we do not know the IP address of the other end of the tunnel beforehand. In the example, Router 2 only needs to know the host name to create the IPsec tunnel, it doesn't need to know the IP addresses of the routers that are going to connect.

Using the *TEMPLATE 1 IKE IDTYPE FQDN* command, we can tell the router to authenticate using the host name rather than the IP address (which is the default option).

```
GAS1 Config>protocol ip
-- Internet protocol user configuration --
GAS1 IP config>ipsec
-- IPSec user configuration --
GAS1 IPSec config>enable
GAS1 IPSec config>template 1 ?
  manual     Manual template
  isakmp     Isakmp template
  dynamic    Dynamic template
GAS1 IPSec config>template 1 isakmp ?
  des       Encryption algorithm DES (Data Encryption Standard)
  tdes      Encryption algorithm TDES (Triple Data Encryption Standard)
  aes128    Encryption algorithm AES using 128-bit key (Advanced Encryption
            Standard)
  aes192    Encryption algorithm AES using 192-bit key (Advanced Encryption
            Standard)
  aes256    Encryption algorithm AES using 256-bit key (Advanced Encryption
            Standard)
GAS1 IPSec config>template 1 isakmp des ?
  md5       Authentication algorithm MD5
  sha1      Authentication algorithm SHA1
  sha256    Authentication algorithm SHA256
  sha384    Authentication algorithm SHA384
  sha512    Authentication algorithm SHA512
GAS1 IPSec config>template 1 isakmp des md5
GAS1 IPSec config>template 1 ?
  isakmp                 Isakmp template
  destination-address    Address of the other remote end of the tunnel
  discover               Use TED to discover the remote end of the tunnel
  backup-destination     Backup destination IP address
  disable-fast-return-from-backup    Disable checking of Main availability and
                                     fast return to Main from Backup
  udp-encapsulation      Enables UDP encapsulation
  udp-ike                Enables IKE UDP encapsulation
  life                   Introduces the SAs life span created from the
                         template
  ike                    Configures parameters relative to the IPSec IKE mode
  keepalive              Enables the available keepalive services
  config                 Isakmp configuration
  aggressive             Aggressive configuration mode ciphered/clear
  send-original-pkt      Send original packet after tunnel establishment
  set-label              Set Label Value of IKE and IPSEC packets
  no                     Negates a command or sets its default
GAS1 IPSec config>template 1 destination-address ?
  <a.b.c.d>      Ipv4 format
  <word>         Text
```

```
GAS1 IPSec config>template 1 destination-address 192.60.1.21
GAS1 IPSec config>template 1 ike ?
  mode                    Mode in which phase I of the ISAKMP/IKE exchange is
                          carried out
  method                  Establishes the authentication method used by the
                          device
  id                      Identifier used during phase 1 of the ISAKMP/IKE
                          exchange
  idtype                  Types of identifiers used during phase 1 of the
                          ISAKMP/IKE exchange
  group                   group
  fragmentation           IKE Fragmentation
  lifetime-negotiation    Enables lifetime negotiation
  early-retry             Retry IKE negotiation in 1/4 of purgetime
  natt-version            Send natt vendor id specific version
  no                      Negates a command or set it default
GAS1 IPSec config>template 1 ike mode ?
  aggressive    Aggressive mode
  main          Main mode
GAS1 IPSec config>template 1 ike mode aggressive
GAS1 IPSec config>template 1 ike idtype ?
  ip        IP Address
  fqdn      FQDN
  ufqdn     UFQDN
  keyid     keyid
  asn-dn    asn-dn
GAS1 IPSec config>template 1 ike idtype fqdn
GAS1 IPSec config>
```

The resulting ISAMKP template configuration is:

```
GAS1 IPSec config>show config
; Showing Menu and Submenus Configuration for access-level 15 ...

         enable
;
         template 1 isakmp des md5
         template 1 destination-address 192.60.1.21
         template 1 ike mode aggressive
         template 1 ike natt-version rfc
         template 1 ike idtype fqdn
         template 1 keepalive dpd
;
         advanced dpd no always-send
GAS1 IPSec config>
```

Having created the ISAKMP template, we can create the DYNAMIC template.

The first thing we need to do is define the type of service, ESP or AH. ESP provides authentication, data integrity, source address authentication and anti-replay protection. AH doesn't provide confidentiality. Next, we need to define a specific encryption (DES) and authentication (SHA1) type, as indicated in the initial security specifications.

The dynamic and ISAKMP templates should have a different template ID. In this particular example, the dynamic template ID is 2.

As with the ISAKMP template, we again need to specify the destination address, but we also have to indicate the source address, i.e., the address of our ADSL interface. We have also enabled the KEEPALIVE option on this template to ensure that the device at the other end of the tunnel keeps its SA open.

```
GAS1 IPSec config>template 2 dynamic ?
  esp               ESP security service (Encapsulating Security Payload)
  ah                AH security service (Authentication Header)
  gdoi-ks-policy    Encryption and authentication algorithms received from
                    GDOI Key Server
GAS1 IPSec config>template 2 dynamic esp ?
  des       Encryption algorithm DES (Data Encryption Standard)
  tdes      Encryption algorithm TDES (Triple Data Encryption Standard)
  aes128    Encryption algorithm AES using 128-bit key (Advanced Encryption
            Standard)
```

```
  aes192    Encryption algorithm AES using 192-bit key (Advanced Encryption
            Standard)
  aes256    Encryption algorithm AES using 256-bit key (Advanced Encryption
            Standard)
GAS1 IPSec config>template 2 dynamic esp des ?
  md5       Authentication algorithm MD5
  sha1      Authentication algorithm SHA1
  sha256    Authentication algorithm SHA256
  sha384    Authentication algorithm SHA384
  sha512    Authentication algorithm SHA512
  none      No authentication algorithm
GAS1 IPSec config>template 2 dynamic esp des sha1
GAS1 IPSec config>template 2 source-address ?
  <a.b.c.d>      Ipv4 format
  <interface>    Interface name
GAS1 IPSec config>template 2 source-address 192.60.1.24
GAS1 IPSec config>template 2 destination-address ?
  <a.b.c.d>      Ipv4 format
  <word>         Text
GAS1 IPSec config>template 2 destination-address 192.60.1.21
GAS1 IPSec config>template 2 keepalive ?
  keepalive    Enables the available keepalive services
  no           Negates a command or sets its default
GAS1 IPSec config>template 2 keepalive keepalive
GAS1 IPSec config>
```

The resulting ISAKMP and DYNAMIC template configuration is:

```
GAS1 IPSec config>show config
; Showing Menu and Submenus Configuration for access-level 15 ...

         enable
;
         template 1 isakmp des md5
         template 1 destination-address 192.60.1.21
         template 1 ike mode aggressive
         template 1 ike natt-version rfc
         template 1 ike idtype fqdn
         template 1 keepalive dpd
;
         template 2 dynamic esp des sha1
         template 2 source-address 192.60.1.24
         template 2 destination-address 192.60.1.21
         template 2 keepalive keepalive
;
         advanced dpd no always-send
GAS1 IPSec config>
```

The last thing we need to do is to configure the pre-shared key. This key is the same at both ends of the tunnel.

When entering the key, we have to specify that it is a pre-shared key and that we will be entering a name rather than an IP address (as explained above).

The name we enter is the **domain name** of the other end of the tunnel.

As well as configuring the device host name, we can also configure the device domain. This is done in the following way:

```
GAS1 IP config>dns-domain-name ?
  <word>    Text
GAS1 IP config>dns-domain-name madrid.es
Domain name : madrid.es
Domain Name configured.
GAS1 IP config>
```

The domain name has not been used in this example. If we try to display the domain name, we will be told that it hasn't been configured and that the name "**GAS1.** " will be used instead. We need to configure this name on the device at the other end of the tunnel (i.e., on Router 2) when specifying the common pre-shared keys.

```
GAS1 IP config>list dns-domain-name
```

```
No Domain Name configured.
Partial DNS name : GAS1.
```

Since the domain name has not been configured on Router 2, we need to enter "**HOST.**" as the host name to use in the key on Router 1. **HOST** is the only host name configured on the device.

```
GAS1 IPSec config>key preshared hostname HOST. plain 1234567890123456
```

2.4.3.1.4 Creating SDPs

Finally, we need to establish the SPDs. That is, we need to assign an ACL to one of the templates we've created. In the following example, the generic list that we have configured and that will be associated with a template, is access list 102. The template that will be associated with it is the dynamic template, i.e., ID 2.

```
GAS1 IPSec config>assign-access-list ?
  <100..1999>    Enter extended access list id
GAS1 IPSec config>assign-access-list 102
GAS1 IPSec config>map-template ?
  <100..1999>    Enter extended access list id
GAS1 IPSec config>map-template 102 ?
  <1..65535>    Enter template id(1-65534)
GAS1 IPSec config>map-template 102 2
GAS1 IPSec config>
```

The IPsec configuration for Router 1 looks like this:

```
GAS1 IPSec config>list all


IPSec Access Control.
Access Control is: enabled
QOS Preclassify is: disabled

Access Lists assigned to IPSec:
  Extended Access List 102
  Templates: 2


Extended Access List 102, assigned to IPSec

1     PERMIT  SRC=192.60.1.24/32  DES=172.24.0.0/16  Conn:0

TEMPLATES
1 isakmp DES MD5  SRC=internal DES=192.60.1.21
   LifeTime:1h0m0s
   IKE AGGRESSIVE
   PRESHARED
   fqdn ID TYPE
   OAKLEY GROUP 1
   DPD enabled

2 dynamic ESP-DES ESP-SHA1  SRC=192.60.1.24  DES=192.60.1.21
   LifeTime:1h0m0s
   PFS disabled
   Keep Alive enabled

1 key entries
   HOST. ************
0 rsakey entries
Id.           Date.         Len         Certificate Authority (CA)           Cert sn.

Period to check LDAP servers not configured. Using default value: 24h0m0s

KeepAlive Configuration:
   Maximum number of encoded packets without receiving an answer: 0.
   Timeout after last packet encoded: 0 seconds.

DPD Configuration:
Idle period(secs) before sending DPD keepalives: 60
```

```
Maximum number of DPD keepalives not acknowledged: 3
Period of time(secs) between DPD keepalives: 5
Always send keepalive after idle period expiration : DISABLED
Anti-replay : DISABLED

Hash Configuration:
Maximum number of entries in hash table: 50000
Shift constant for exponential moving average calculation: 8

Check-out time (%) - from SA's end-lifetime - to renegotiate : 10

SA's purge timeout:  15

Period of connected SA's notification event:  0

NAT Translation Port:  4500

Use software exponentiation

Maximum number of IPSec headers in a packet:  1

GAS1 IPSec config>
```

The *SHOW CONFIG* command provides the following information:

```
GAS1 IPSec config>show config
; Showing Menu and Submenus Configuration for access-level 15 ...

         enable
         assign-access-list 102
;
         template 1 isakmp des md5
         template 1 destination-address 192.60.1.21
         template 1 ike mode aggressive
         template 1 ike natt-version rfc
         template 1 ike idtype fqdn
         template 1 keepalive dpd
;
         template 2 dynamic esp des sha1
         template 2 source-address 192.60.1.24
         template 2 destination-address 192.60.1.21
         template 2 keepalive keepalive
;
         map-template 102 2
         key preshared hostname HOST. ciphered 0xE21C47018BC8B868FB72F48DC4363FC0CF
ABF60C9FFE0286
         advanced dpd no always-send
GAS1 IPSec config>
```

2.4.3.2 Configuring Router 2

2.4.3.2.1 Configuring the host name, IP addresses and rules

The host name and IP protocol parameter settings are similar to those in Router 1.

```
bintec               (c)1996-2002
*PROCESS 4
User Configuration
Config>set hostname HOST
```

When configuring the IP protocol, we must make sure we configure the correct interface addresses, as the ethernet0/0 interface connects the network card to LAN 172.24.0.0. We also need to assign the IP address to the ADSL interface with which you are going to set up the tunnel.

```
HOST config>network atm0/0

; -- ATM interface configuration --
HOST atm0/0 config>ip address 192.60.1.24 255.255.255.0
HOST atm0/0 config>exit
```

```
HOST config>network ethernet0/0

-- Ethernet Interface User Configuration --
HOST ethernet0/ config>ip address 172.24.78.101 255.255.0.0
```

2.4.3.2.2 Creating the access control lists

Having configured all the IP parameters, we can now turn to the IPsec configuration itself.

While the access control list (ACL) settings are similar to those on Router 1, we should take care when configuring the source and destination IP addresses.

```
HOST Access Lists config>show config
; Showing Menu and Submenus Configuration for access-level 15 ...

      access-list 103
         entry 1 default
         entry 1 permit
         entry 1 source address 172.24.0.0 255.255.0.0
         entry 1 destination address 192.60.1.24 255.255.255.255
;
      exit
;
HOST Access Lists config>
```

2.4.3.2.3 Creating templates

As with Router 1, the negotiation mode in the ISAKMP and dynamic templates is aggressive mode. We need to use the same pre-shared key that was configured on Router 1 but this time we need to specify that the key corresponds to the "**GAS1**" *hostname*.

When creating the ISAKMP template, we need to specify the type of encryption (DES) and authentication (MD5) to use (as indicated in the initial security specifications). These should match the algorithms configured on Router 1.

When we create the template, we have to specify an ID number that will be used throughout the template configuration. We also have to specify the tunnel destination IP to which we are connecting; since we don't know the IP address of the device that is going to connect to Router 2 (we only know the host name), the destination IP address will be 0.0.0.0. We must also specify the use of aggressive mode, and that the IDTYPE will be FQDN. This ensures that the host name is used for authentication purposes rather than the IP address which is the default option.

```
HOST IPSec config>show config
; Showing Menu and Submenus Configuration for access-level 15 ...

         enable
;
         template 1 isakmp des md5
         template 1 ike mode aggressive
         template 1 ike natt-version rfc
         template 1 ike idtype fqdn
         template 1 keepalive dpd
;
         advanced dpd no always-send
HOST IPSec config>
```

Having created the ISAKMP template, we can go about creating the DYNAMIC template with ESP, DES encryption and SHA1 authentication (as we did for Router 1). To ensure that the DYNAMIC template configuration doesn't overwrite the ISAKMP template configuration, the DYNAMIC and ISAKMP templates should have a different template ID. In the example, the template ID is 2.

As with the ISAKMP template, we again need to specify the **destination address** (**0.0.0.0**), but we also need to specify the source address, i.e., the address of our ADSL interface. The KEEPALIVE option is not enabled on this template. This frees up processing time for Router 2 and ensures that the connecting routers are responsible for checking the SA is open.

The resulting ISAKMP and DYNAMIC template configuration is:

```
HOST IPSec config>show config

         enable
;
         template 1 isakmp des md5
```

```
        template 1 ike mode aggressive
        template 1 ike natt-version rfc
        template 1 ike idtype fqdn
        template 1 keepalive dpd
;
        template 2 dynamic esp des sha1
        template 2 source-address 192.60.1.21
        template 2 life duration seconds 30m
;
        advanced dpd no always-send
HOST IPSec config>
```

Finally, we must configure the pre-shared key. This key is the same at both ends of the tunnel.

When entering the key, we must specify that it is a pre-shared key and that we will be entering a name rather than an IP address (as explained above).

The name we enter is the remote tunnel endpoint's **domain name** (as explained in the case of Router 1).

In this particular example, we must use the name "**GAS1.** ", which is Router 1's domain name.

```
HOST IPSec config>key preshared hostname GAS1. plain 1234567890123456
HOST IPSec config>
```

If additional routers (i.e., not just Router 1) are to be connected to this router, we must specify a host name and the corresponding key for each of them.

2.4.3.2.4 Creating SPDs

Finally, we must establish the *SPDs*. That is, we need to assign an ACL to one of the templates we've created. In the following example, the extended generic list that we have configured and that will be assigned to IPsec and associated with a template, is access-list 103. The template that will be associated is the dynamic template, ID 2.

```
HOST IPSec config>assign-access-list 103
HOST IPSec config>map-template 103 2
HOST IPSec config>
```

Finally, we can free up more processing time for Router 2 in two ways: firstly, by specifying not to re-negotiate the SA when the specified lifetime percentage has been reached, and secondly, by specifying that the other end of the tunnel (Router 1) will re-negotiate the SA.

```
HOST IPSec config>advanced renegotiation-time 0
HOST IPSec config>
```

The resulting IPsec configuration is:

```
HOST IPSec config>list all


IPSec Access Control.
Access Control is: enabled
QOS Preclassify is: disabled

Access Lists assigned to IPSec:
  Extended Access List 103
  Templates: 2


Extended Access List 103, assigned to IPSec

1     PERMIT  SRC=172.24.0.0/16  DES=192.60.1.24/32  Conn:0

TEMPLATES
1 isakmp DES MD5  SRC=internal DES=0.0.0.0
   LifeTime:1h0m0s
   IKE AGGRESSIVE
   PRESHARED
   fqdn ID TYPE
   OAKLEY GROUP 1
   DPD enabled
```

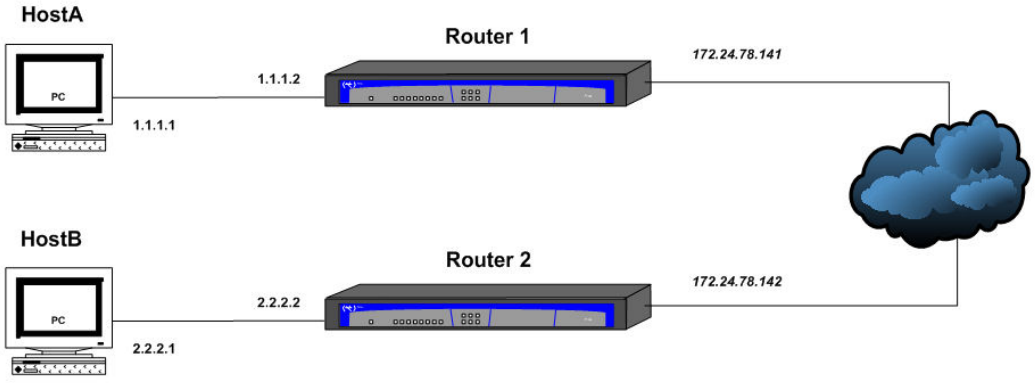
```
2 dynamic ESP-DES ESP-SHA1  SRC=192.60.1.21  DES=0.0.0.0
   LifeTime:0h30m0s
   PFS disabled

1 key entries
   GAS1. ************
0 rsakey entries
Id.           Date.         Len         Certificate Authority (CA)           Cert sn.

Period to check LDAP servers not configured. Using default value: 24h0m0s

KeepAlive Configuration:
   Maximum number of encoded packets without receiving an answer: 0.
   Timeout after last packet encoded: 0 seconds.

DPD Configuration:
Idle period(secs) before sending DPD keepalives: 60
Maximum number of DPD keepalives not acknowledged: 3
Period of time(secs) between DPD keepalives: 5
Always send keepalive after idle period expiration : DISABLED
Anti-replay : DISABLED

Hash Configuration:
Maximum number of entries in hash table: 50000
Shift constant for exponential moving average calculation: 8

Check-out time (%) - from SA's end-lifetime - to renegotiate : 0

SA's purge timeout:  15

Period of connected SA's notification event:  0

NAT Translation Port:  4500

Use software exponentiation

Maximum number of IPSec headers in a packet:  1

HOST IPSec confi>
```

Running the *SHOW CONFIG* command will give us the following:

```
HOST IPSec config>show config
; Showing Menu and Submenus Configuration for access-level 15 ...

         enable
         assign-access-list 103
;
         template 1 isakmp des md5
         template 1 ike mode aggressive
         template 1 ike natt-version rfc
         template 1 ike idtype fqdn
         template 1 keepalive dpd
;
         template 2 dynamic esp des sha1
         template 2 source-address 192.60.1.21
         template 2 life duration seconds 30m
;
         map-template 103 2
         key preshared hostname GAS1. ciphered 0xE21C47018BC8B868FB72F48DC4363FC0CF
ABF60C9FFE0286
         advanced renegotiation-time 0
         advanced dpd no always-send
HOST IPSec config>
```

The routers can communicate securely, and the pre-shared key is the only key to protect in this case.

2.4.4 Example 4: Tunnel End-point Discovery



This scenario shows how to use Tunnel End-point Discovery (TED) in dynamic IPsec (IPsec IKE). Two routers have been configured to open an IPsec tunnel to one another in the same way as described in example 2. In this case, however, neither of them have been given the remote peer's IP address needed to open the tunnel. When one of the hosts protected by a router wishes to communicate with its remote endpoint (e.g., host A with host B), the router uses TED to find its colleague and initiate ISAKMP negotiations.

2.4.4.1 Configuring Router 1

2.4.4.1.1 Configuring the hostname, addresses and IP rules

The IP configuration for this example will be very basic and consists of the addresses of the two networks connected by the router and a route to reach the network protected by the remote end:

```
*p 4
Config>network ethernet0/0

-- Ethernet Interface User Configuration --
ethernet0/0 config>ip address 172.24.78.141 255.255.0.0
ethernet0/0 config>ip address 1.1.1.2 255.255.255.0 secondary
ethernet0/0 config>exit
Config>protocol ip

-- Internet protocol user configuration --
IP config>route 2.2.2.0 255.255.255.0 172.24.78.142
IP config>exit
Config>
```

2.4.4.1.2 Creating the access control lists

Having configured the IP parameters, we now need to configure the ACLs. This we do by accessing the generic lists setup menu, picking a number from the list that corresponds to an extended list (between 100 and 199 - we have selected 101 in our example), indicating an entry ID from the list (in this case 1), and entering the desired values for the following parameters:

- The source address of the packets we want to collide with the access list; in this case the subnet with the clients that we will protect.
- The destination address of the packets that are going to collide, in this case the subnet with the clients that the other router protects.
- The action to be taken on the packets, in this case IPsec processing (**PERMIT**).

```
Config>feature access-lists

-- Access Lists user configuration --
Access Lists config>access-list 101


Extended Access List 101>entry 1 default
Extended Access List 101>entry 1 permit
Extended Access List 101>entry 1 source address 1.1.1.0 255.255.255.0
Extended Access List 101>entry 1 destination address 2.2.2.0 255.255.255.240
Extended Access List 101>exit
```

```
Access Lists config>exit
Config>
```

2.4.4.1.3 Creating templates

Now, we need to create the ISAKMP and dynamic templates. When creating the ISAKMP template, we can use the *DISCOVER* option to specify the use of TED for discovering the remote tunnel end. When creating the dynamic template, we must be careful not to specify the remote address of the tunnel; this is because we don't know it yet and it will be discovered during the TED process.

```
Config>protocol ip
-- Internet protocol user configuration --
IP config>ipsec
-- IPSec user configuration --
IPSec config>enable
IPSec config>template 1 isakmp tdes sha1
IPSec config>template 1 discover
IPSec config>template 1 life duration seconds 45m
IPSec config>template 3 dynamic esp tdes md5
IPSec config>template 3 source-address 172.24.78.141
IPSec config>template 3 life type both
IPSec config>template 3 life duration seconds 45m
IPSec config>key preshared ip 0.0.0.0 ciphered 0xD8599397F3F05E04A00A56234D376BCD
IPSec config>event address-filter 0.0.0.0 0.0.0.0
```

2.4.4.1.4 Creating the SPDs

Finally, we need to establish the SPDs. That is, we need to assign an ACL to one of the templates we've created. In the following example, the extended access list that we have configured and that must be assigned to IPsec and associated with a template, is access list 101. The template that must be associated is the dynamic template, ID 3.

```
IPSec config>assign-access-list 101
IPSec config>map-template 101 3
```

The IPsec configuration will be as follows:

```
IPSec config>list all


IPSec Access Control.
Access Control is: enabled
QOS Preclassify is: disabled

Access Lists assigned to IPSec:
  Extended Access List 101
  Templates: 3


Extended Access List 101, assigned to IPSec

1     PERMIT  SRC=1.1.1.0/24  DES=2.2.2.0/28  Conn:0

TEMPLATES
1 isakmp 3DES SHA1  SRC=internal DES=0.0.0.0
   LifeTime:0h45m0s
   IKE MAIN
   PRESHARED
   addr4 ID TYPE
   OAKLEY GROUP 1
   DPD enabled
   Tunnel End-point Discovery enabled

3 dynamic ESP-3DES ESP-MD5  SRC=172.24.78.141  DES=0.0.0.0
   LifeTime:0h45m0s 4608000 kbytes
   PFS disabled

1 key entries
   0.0.0.0 ************
0 rsakey entries
```

```
Id.           Date.         Len         Certificate Authority (CA)           Cert sn.

Period to check LDAP servers not configured. Using default value: 24h0m0s

KeepAlive Configuration:
   Maximum number of encoded packets without receiving an answer: 0.
   Timeout after last packet encoded: 0 seconds.

DPD Configuration:
Idle period(secs) before sending DPD keepalives: 60
Maximum number of DPD keepalives not acknowledged: 3
Period of time(secs) between DPD keepalives: 5
Always send keepalive after idle period expiration : DISABLED
Anti-replay : DISABLED

Hash Configuration:
Maximum number of entries in hash table: 50000
Shift constant for exponential moving average calculation: 8

Check-out time (%) - from SA's end-lifetime - to renegotiate : 10

SA's purge timeout:  15

Period of connected SA's notification event:  0

NAT Translation Port:  4500

Use software exponentiation

Maximum number of IPSec headers in a packet:  1

IPSec config>
```

And when we issue the *SHOW CONFIG* command, the complete device configuration is:

```
Config>show config
; Showing Menu and Submenus Configuration for access-level 15 ...

   log-command-errors
   no configuration
   set data-link sync serial0/0
   set data-link sync serial0/1
   feature access-lists
; -- Access Lists user configuration --
      access-list 101
         entry 1 default
         entry 1 permit
         entry 1 source address 1.1.1.0 255.255.255.0
         entry 1 destination address 2.2.2.0 255.255.255.240
;
      exit
;
   exit
;
;
   network ethernet0/0
; -- Ethernet Interface User Configuration --
      ip address 172.24.78.141 255.255.0.0
      ip address 1.1.1.2 255.255.255.0 secondary
;
   exit
;
;
   protocol ip
; -- Internet protocol user configuration --
      route 2.2.2.0 255.255.255.0 172.24.78.142
;
      ipsec
```

```
; -- IPSec user configuration --
        enable
        assign-access-list 101
;
        template 1 isakmp tdes sha1
        template 1 discover
        template 1 life duration seconds 45m
        template 1 ike natt-version rfc
        template 1 keepalive dpd
;
        template 3 dynamic esp tdes md5
        template 3 source-address 172.24.78.141
        template 3 life type both
        template 3 life duration seconds 45m
;
        map-template 101 3
        key preshared ip 0.0.0.0 ciphered 0xD8599397F3F05E04A00A56234D376BCD
        advanced dpd no always-send
      exit
;
   exit
;
;
   dump-command-errors
   end
Config>
```

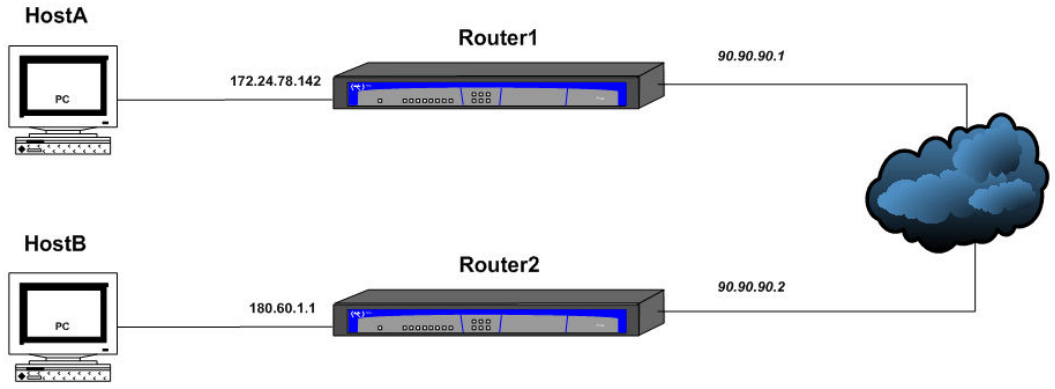
2.4.4.2 Configuring Router 2

The configuration for Router 2 is similar to that of Router 1, with the only difference being that the source and destination addresses on the ACLs and templates are the other way round. Thus the configuration for Router 2 would look like this:

```
Config>show config
; Showing Menu and Submenus Configuration for access-level 15 ...

   log-command-errors
   no configuration
   set data-link sync serial0/0
   set data-link sync serial0/1
   feature access-lists
; -- Access Lists user configuration --
      access-list 101
         entry 1 default
         entry 1 permit
         entry 1 source address 2.2.2.0 255.255.255.240
         entry 1 destination address 1.1.1.0 255.255.255.0
;
      exit
;
   exit
;
;
   network ethernet0/0
; -- Ethernet Interface User Configuration --
      ip address 172.24.78.142 255.255.0.0
      ip address 2.2.2.2 255.255.255.0 secondary
;
   exit
;
;
   protocol ip
; -- Internet protocol user configuration --
      route 1.1.1.0 255.255.255.0 172.24.78.141
;
      ipsec
; -- IPSec user configuration --
```

```
         enable
         assign-access-list 101
;
         template 1 isakmp tdes sha1
         template 1 discover
         template 1 life duration seconds 4m
         template 1 ike natt-version rfc
         template 1 keepalive dpd
;
         template 3 dynamic esp tdes md5
         template 3 source-address 172.24.78.142
         template 3 life type both
         template 3 life duration seconds 6m
;
         map-template 101 3
         key preshared ip 0.0.0.0 ciphered 0xD8599397F3F05E04A00A56234D376BCD
         advanced dpd no always-send
      exit
;
   exit
;
;
   dump-command-errors
   end
Config>
```

2.4.5 Example 5: Permanent tunnel



This scenario shows how to configure two devices to create a permanent tunnel to one another. The configuration we use in this particular example will be very similar to the one we used in example 2 (dynamic IKE) but we will add the necessary commands to ensure the tunnel is always open.

2.4.5.1 Configuring Router 1

2.4.5.1.1 Configuring IP, Lca, templates and SPDs

We start with a basic dynamic IPsec configuration with two routers protecting both subnets. The *ADVANCED RENEGOTIATION-TIME 100* command is worth mentioning here because it allows the tunnel to renegotiate even when there isn't any traffic.

```
Config>show config
; Showing Menu and Submenus Configuration for access-level 15 ...

   log-command-errors
   no configuration
   set data-link sync serial0/0
   set data-link sync serial0/1
   feature access-lists
; -- Access Lists user configuration --
      access-list 101
         entry 1 default
         entry 1 permit
         entry 1 source address 172.24.0.0 255.255.0.0
```

```
         entry 1 destination address 180.60.0.0 255.255.0.0
;
      exit
;
   exit
;
;
   network ethernet0/0
; -- Ethernet Interface User Configuration --
      ip address 172.24.78.142 255.255.0.0
      ip address 90.90.90.1 255.255.255.0 secondary
;
   exit
;
;
   protocol ip
; -- Internet protocol user configuration --
      route 0.0.0.0 0.0.0.0 90.90.90.2
;
      ipsec
; -- IPSec user configuration --
         enable
         assign-access-list 101
;
         template 1 isakmp tdes sha1
         template 1 destination-address 90.90.90.2
         template 1 ike natt-version rfc
         template 1 keepalive dpd
;
         template 3 dynamic esp tdes md5
         template 3 source-address 90.90.90.1
         template 3 destination-address 90.90.90.2
;
         map-template 101 3
         key preshared ip 90.90.90.2 ciphered 0xD8599397F3F05E04A00A56234D376BCD
         advanced renegotiation-time 100
         advanced dpd no always-send
      exit
;
   exit
;
;
   dump-command-errors
   end
Config>
```

We will also be using the proprietary TIDP protocol to ensure the tunnel has traffic and can be opened if the device reboots or if there isn't any communication between the tunnel ends when the renegotiation is taking place. We shall configure the device to periodically send a discovery packet whose source and destination IPs collide with the access list used in IPsec. The device in the example has been configured to send a packet every two minutes (this is more than enough time to open the tunnel when the device reboots).

```
Config>feature ip-discovery
-- IP Discovery Protocol configuration --
TIDP config>discovery-station 1 ip 180.60.1.1
TIDP config>discovery-station 1 source ip 172.24.78.142
TIDP config>discovery-station 1 timer 2m
TIDP config>exit
Config>
```

2.4.5.2 Configuring Router 2

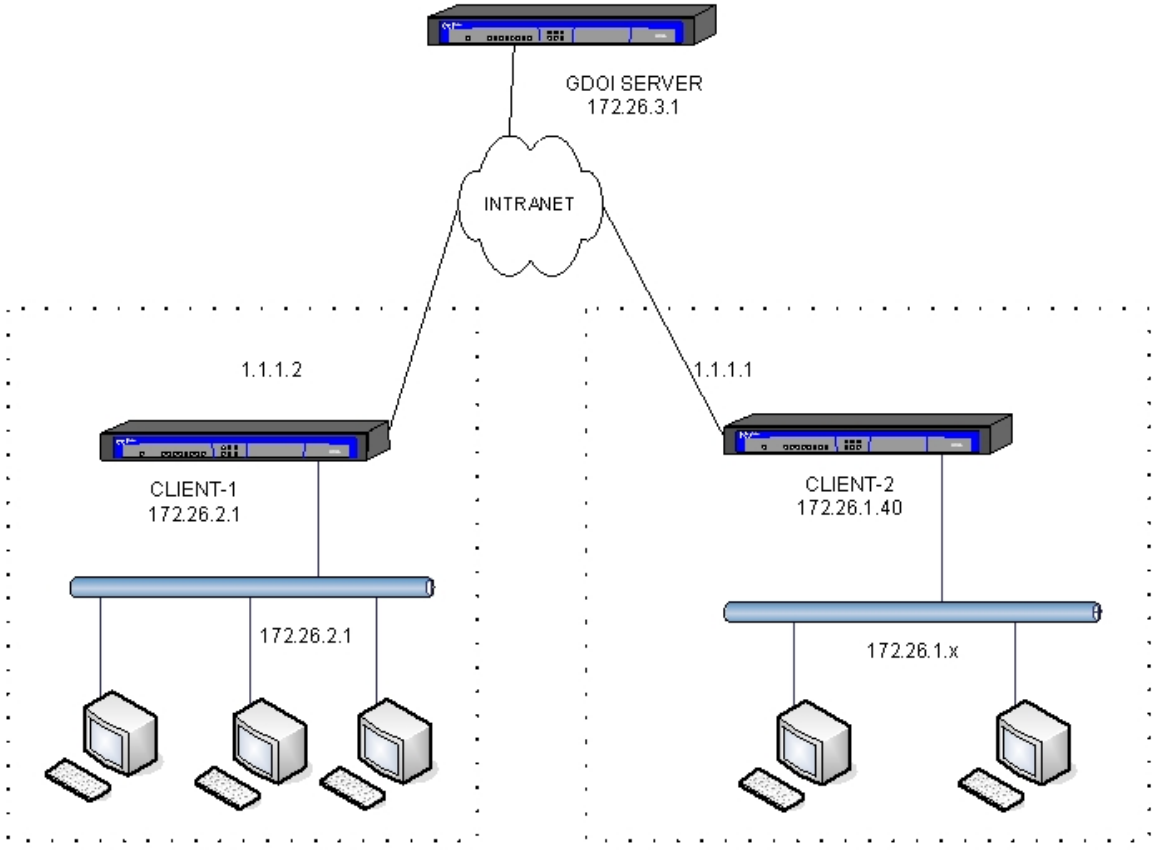
The configuration for Router 2 is similar to the one for Router 1, with the only differences being that the source and destination addresses on the access lists and templates are the other way round and we don't need to configure the renegotiation time (the other end is going to open the tunnel) or the TIDP interval. The configuration for Router 2 would therefore look like this:

```
Config>show config
```

```
; Showing Menu and Submenus Configuration for access-level 15 ...

   log-command-errors
   no configuration
   set data-link sync serial0/0
   set data-link sync serial0/1
   feature access-lists
; -- Access Lists user configuration --
      access-list 101
         entry 1 default
         entry 1 permit
         entry 1 source address 180.60.0.0 255.255.0.0
         entry 1 destination address 172.24.0.0 255.255.0.0
;
      exit
;
   exit
;
;
   network ethernet0/0
; -- Ethernet Interface User Configuration --
      ip address 180.60.1.1 255.255.0.0
      ip address 90.90.90.2 255.255.255.0 secondary
;
   exit
;
;
;
   protocol ip
; -- Internet protocol user configuration --
      route 0.0.0.0 0.0.0.0 90.90.90.1
;
      ipsec
; -- IPSec user configuration --
         enable
         assign-access-list 101
;
         template 1 isakmp tdes sha1
         template 1 destination-address 90.90.90.1
         template 1 ike natt-version rfc
         template 1 keepalive dpd
;
         template 3 dynamic esp tdes md5
         template 3 source-address 90.90.90.2
         template 3 destination-address 90.90.90.1
;
         map-template 101 3
         key preshared ip 90.90.90.1 ciphered 0xD8599397F3F05E04A00A56234D376BCD
         advanced dpd no always-send
      exit
;
   exit
;
;
;
   dump-command-errors
   end
Config>
```

2.4.6 Example 6: GDOI

This scenario shows how to configure the devices so that the GDOI protocol is used to negotiate IPsec keys and encryption policies. We want to encrypt traffic between 172.26.1.x and 172.26.2.x using triple DES and SHA1. To do this, we will need to configure a server at a central point in the network and two clients (one at each site).



2.4.6.1 Configuring the server

The server configuration consists of an access list, number 100, that allows all traffic to network 172.26.0.0. This list is assigned to IPsec and subsequently associated with the GDOI group security association in use (SA 1). SA 1 specifies triple DES encryption and SHA1 authentication with key renewal every 5 minutes.

The rekey method is AES 256 bit encryption in unicast mode with key renewal every 10 minutes. Rekey messages are authenticated using a public key, called MYKEY, that we must generate beforehand:

```
IPSec config$key rsa generate MYKEY 512
```

We also need to configure an ISAKMP template for phase I negotiations with GDOI clients, and a pre-shared key to allow the negotiation to take place.

The GDOI server configuration looks like this:

```
Config>show config
; Showing Menu and Submenus Configuration for access-level 15 ...

   log-command-errors
   no configuration
   add device x25 1
   set data-link frame-relay serial0/0
   set data-link sync serial0/1
   feature access-lists
; -- Access Lists user configuration --
      access-list 100
         entry 1 default
         entry 1 permit
         entry 1 source address 172.26.0.0 255.255.0.0
         entry 1 destination address 172.26.0.0 255.255.0.0
;
      exit
;
   exit
;
;
```

```
   network ethernet0/0
; -- Ethernet Interface User Configuration --
      ip address 172.26.3.1 255.255.255.0
;
   exit
;
;
   protocol ip
; -- Internet protocol user configuration --
      internal-ip-address 172.26.3.1
;
      route 0.0.0.0 0.0.0.0 172.26.3.2
;
      ipsec
; -- IPSec user configuration --
         enable
         assign-access-list 100
;
         template 1 isakmp tdes md5
         template 1 ike natt-version rfc
         template 1 keepalive no dpd
;
         key preshared ip 172.26.2.1 ciphered 0x0DD598B4F74E201E
         key preshared ip 172.26.1.40 ciphered 0x0DD598B4F74E201E
         advanced dpd no always-send
         gdoi group 2
; -- GDOI user configuration --
            identity number 2
            rekey algorithm aes-256
            rekey authentication rsa MYKEY
            rekey lifetime seconds 10m
            rekey retransmit 10s number 3
            rekey transport unicast
            sa ipsec 1
               lifetime 5m
               match address ipv4 100
               transform-set tdes sha1
            exit
;
         exit
;
      exit
;
   exit
;
;
   dump-command-errors
   end
Config>
```

2.4.6.2 Configuring client 1

To provide IP connectivity, we configure the Ethernet and PPP interface addresses according to the diagram at the beginning of this example. The PPP interface IP address is 1.1.1.2 and the Ethernet IP address is 172.26.2.1.

An access list needs to be configured to associate it with the dynamic template. The access list consists of one or more "*deny*" entries for isolating specific traffic on which IPsec will not be applied (even though the downloaded server lists indicate that it should be applied), and a final "*permit*" entry. Given that in this particular example we want to encapsulate all traffic between networks 172.26.1.0/24 and 172.26.2.0/24 without exception, the access list only contains the "*permit*" entry.

An ISAKMP template and a pre-shared key for use in phase I negotiations with the server needs to be configured, as does a dynamic template with the chosen GDOI group, in this case 2. The ISAKMP and dynamic templates must have the GDOI server IP as the destination IP.

The configuration for client 1 is as follows:

```
;Config>show config
; Showing Menu and Submenus Configuration for access-level 15 ...
```

```
   log-command-errors
   no configuration
   add device ppp 1
   set data-link sync serial0/0
   set data-link sync serial0/1
   feature access-lists
; -- Access Lists user configuration --
      access-list 100
         entry 1 default
         entry 1 permit
;
      exit
;
   exit
;
;
   network ethernet0/0
; -- Ethernet Interface User Configuration --
      ip address 172.26.2.1 255.255.255.0
;
   exit
;
   network serial0/0
; -- Interface Synchronous Serial Line. Configuration --
      speed 256000
   exit
;
;
;
   network ppp1
; -- Generic PPP User Configuration --
      ip address 1.1.1.2 255.255.255.0
;
      base-interface
; -- Base Interface Configuration --
         base-interface serial0/0 link
;
      exit
;
   exit
;
;
   protocol ip
; -- Internet protocol user configuration --
      route 172.26.1.0 255.255.255.0 1.1.1.1
;
      ipsec
; -- IPSec user configuration --
         enable
         assign-access-list 100
;
         template 1 dynamic esp tdes sha1
         template 1 source-address 172.26.2.1
         template 1 destination-address 172.26.3.1
         template 1 mapped-to-ifc ppp1
         template 1 gdoi group 2
;
         template 2 isakmp tdes md5
         template 2 destination-address 172.26.3.1
         template 2 ike natt-version rfc
         template 2 keepalive no dpd
;
         map-template 100 1
         key preshared ip 172.26.3.1 ciphered 0x0DD598B4F74E201E
         advanced dpd no always-send
      exit
```

```
;
   exit
;
;
   dump-command-errors
   end
Config>
```

2.4.6.3 Configuring client 2

The configuration for client 2 is similar to the configuration for client 1; all we need to do is to change the IP addresses of the PPP and Ethernet7 interfaces to give us the following:

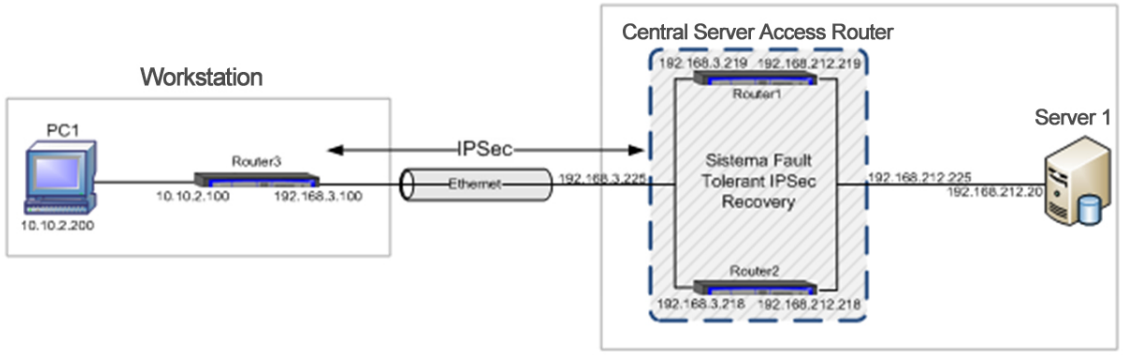
```
Config>show config
; Showing Menu and Submenus Configuration for access-level 15 ...

   log-command-errors
   no configuration
   add device ppp 1
   set data-link sync serial0/0
   set data-link sync serial0/1
   feature access-lists
; -- Access Lists user configuration --
      access-list 100
         entry 1 default
         entry 1 permit
;
      exit
;
   exit
;
;
   network ethernet0/0
; -- Ethernet Interface User Configuration --
      ip address 172.26.1.40 255.255.255.0
;
   exit
;
   network serial0/0
; -- Interface Synchronous Serial Line. Configuration --
      speed 512000
   exit
;
;
;
   network ppp1
; -- Generic PPP User Configuration --
      ip address 1.1.1.1 255.255.255.0
;
      base-interface
; -- Base Interface Configuration --
         base-interface serial0/0 link
;
      exit
;
   exit
;
;
   protocol ip
; -- Internet protocol user configuration --
      internal-ip-address 172.26.1.40
;
      route 172.26.2.0 255.255.255.0 1.1.1.2
;
      ipsec
; -- IPSec user configuration --
         enable
```

```
         assign-access-list 100
;
         template 1 dynamic esp tdes sha1
         template 1 source-address 172.26.1.40
         template 1 destination-address 172.26.3.1
         template 1 mapped-to-ifc ppp1
         template 1 gdoi group 2
;
         template 2 isakmp tdes md5
         template 2 destination-address 172.26.3.1
         template 2 ike natt-version rfc
         template 2 keepalive no dpd
;
         template 3 isakmp tdes md5
         template 3 ike natt-version rfc
         template 3 keepalive no dpd
;
         map-template 100 1
         key preshared ip 172.26.3.1 ciphered 0x0DD598B4F74E201E
         advanced dpd no always-send
      exit
;
   exit
;
;
;
;
   dump-command-errors
   end
Config>
```

2.4.7 Example 7: Fault tolerant IPsec recovery

This scenario shows how to configure fault tolerant IPsec recovery to protect IPsec sessions when the tunnel terminator device fails.

We have a PC in a workstation (PC1) that needs to connect to a central server (Server1). We use a router (Router 3) to establish an IPsec session between the workstation and the central server access router. To make the connection more robust, we have set up fault tolerant IPsec recovery in the central server access router and, in doing so, have converted this router into two routers, Router1 and Router2.



2.4.7.1 Configuring the router in the workstation, Router3

The configuration in Router3 is conceptually similar to the one in example 2, where the tunnel endpoint address is the IP address that the central server access router offers to the outside (i.e., 192.168.3.225). Without further ado, here is the final result:

```
Config>show config
; Showing Menu and Submenus Configuration for access-level 15 ...

   log-command-errors
   no configuration
   set data-link sync serial0/0
   set data-link sync serial0/1
   set hostname Router3
```

```
   feature access-lists
; -- Access Lists user configuration --
      access-list 100
         entry 2 default
         entry 2 permit
         entry 2 source address 10.10.2.0 255.255.255.0
         entry 2 destination address 192.168.212.0 255.255.254.0
;
      exit
;
   exit
;
;
   network ethernet0/0
; -- Ethernet Interface User Configuration --
      ip address 192.168.3.100 255.255.255.0
;
   exit
;
;
   network ethernet0/1
; -- Ethernet Interface User Configuration --
      ip address 10.10.2.100 255.255.255.0
;
   exit
;
;
   protocol ip
; -- Internet protocol user configuration --
      route 0.0.0.0 0.0.0.0 192.168.3.225
;
      ipsec
; -- IPSec user configuration --
         enable
         assign-access-list 100
;
         template 1 isakmp tdes sha1
         template 1 destination-address 192.168.3.225
         template 1 ike natt-version rfc
         template 1 keepalive dpd
;
         template 2 dynamic esp tdes md5
         template 2 destination-address 192.168.3.225
;
         map-template 100 2
         key preshared ip 0.0.0.0 ciphered 0x12D942B46B48645B
         advanced dpd no always-send
      exit
;
   exit
;
;
   dump-command-errors
   end
Config>
```

2.4.7.2 Configuring the central server access router, Router1 and Router2

The central server access router consists of two routers, Router1 and Router2, which have the fault tolerant IPsec recovery system set up between them. For this system, we are required to configure multiple protocols - IPsec, IPsecFT and VRRP - and we will expand on these below.

2.4.7.2.1 Configuring IPsec

The IPsec configuration must be the same on both routers and should not be more complex than the one shown in example 2. We start, therefore, with a similar configuration in IPsec:

```
RouterX Config>show config
[…]
   feature access-lists
; -- Access Lists user configuration --
      access-list 100
         entry 1 default
         entry 1 permit
         entry 1 source address 192.168.212.0 255.255.254.0
         entry 1 destination address 10.10.2.0 255.255.255.0
;
      exit
;
   exit
;
[…]
   protocol ip
; -- Internet protocol user configuration --
      route 10.10.2.0 255.255.255.0 192.168.3.100
;
      ipsec
; -- IPSec user configuration --
         enable
         assign-access-list 100
;
         template 1 isakmp tdes sha1
         template 1 ike natt-version rfc
         template 1 keepalive dpd
         template 1 send-original-pkt
;
         template 2 dynamic esp tdes md5
         template 2 source-address 192.168.3.225
;
         map-template 100 2
         key preshared ip 0.0.0.0 ciphered 0x12D942B46B48645B
         advanced dpd always-send
      exit
;
   exit
;
[…]
RouterX Config>
```

At this point we should focus on one of the configuration parameters in the dynamic templates. This parameter indicates which templates form part of the fault tolerant IPsec recovery system. That is, which of the IPsec sessions remain open should the device that set them up fail. In this particular case, we want to apply fault tolerance to a single dynamic template, template 2.

```
RouterX IPSec config>template 2 fault-tolerant
RouterX IPSec config>
```

The resulting configuration for both Router1 and Router2 is as follows:

```
RouterX Config>show config
[…]
   feature access-lists
; -- Access Lists user configuration --
      access-list 100
         entry 1 default
         entry 1 permit
         entry 1 source address 192.168.212.0 255.255.254.0
         entry 1 destination address 10.10.2.0 255.255.255.0
;
      exit
;
   exit
;
[…]
   protocol ip
; -- Internet protocol user configuration --
```

```
      route 10.10.2.0 255.255.255.0 192.168.3.100
;
      ipsec
; -- IPSec user configuration --
         enable
         assign-access-list 100
;
         template 1 isakmp tdes sha1
         template 1 ike natt-version rfc
         template 1 keepalive dpd
         template 1 send-original-pkt
;
         template 2 dynamic esp tdes md5
         template 2 source-address 192.168.3.225
         template 2 fault-tolerant
;
         map-template 100 2
         key preshared ip 0.0.0.0 ciphered 0x12D942B46B48645B
         advanced dpd always-send
      exit
;
   exit
[…]
RouterX Config>
```

2.4.7.2.2 Configuring IPsecFT

The IPsecFT protocol is responsible for maintaining a database of all IPsec sessions in order to allow a peer to take control of those sessions should one of the routers fail.

To configure IPsecFT, we must first decide which of the devices is going to be the master and which the slave. Our decision will not affect the way the system operates. In our example, Router1 will be the master.

IPsecFT in Router1

Starting with Router1, the first thing we need to do is to access the IPsecFT submenu.

```
Router1 Config>protocol ip
-- Internet protocol user configuration --
Router1 IP config>ipsec
-- IPSec user configuration --
Router1 IPSec config>fault-tolerant
-- Fault tolerant IPSec recovery user configuration --
Router1 IPSecFT config>
```

This subsystem is in slave mode and disabled by default. We need to change the mode so that the router can act as master. Before we enable the mode, there are a couple of required parameters that must be configured if we want it to work properly. These parameters are the IP address of the slave device and the packet source. In this particular case, the slave address is 192.168.212.218 and the source is IP 192.168.212.219.

At this point, we should stress that the source and destination address in IPsecFT must not match the virtual IP address managed in VRRP.

```
Router1 IPSecFT config>mode master
Router1 IPSecFT config>slave-address 192.168.212.218
Router1 IPSecFT config>source-address 192.168.212.219
Router1 IPSecFT config>enable
Router1 IPSecFT config>show menu
; Showing Menu Configuration for access-level 15 ...
            mode master
            slave-address 192.168.212.218
            source-address 192.168.212.219
            enable
Router1 IPSecFT config>
```

Other parameters, such as the port, are optional. In this particular example, these parameters will be set to their default values.

IPsecFT in Router2

Turning now to Router2, we need to access the IPsecFT submenu.

```
Router2 Config>protocol ip
-- Internet protocol user configuration --
Router2 IP config>ipsec
-- IPSec user configuration --
Router2 IPSec config>fault-tolerant
-- Fault tolerant IPSec recovery user configuration --
Router2 IPSecFT config>
```

This device will be in slave mode so there is no need to configure anything. The only parameter that can be configured is the listening port, but we will leave this parameter set to its default value (as we did in the master device). All we need to do at this point is to enable the protocol.

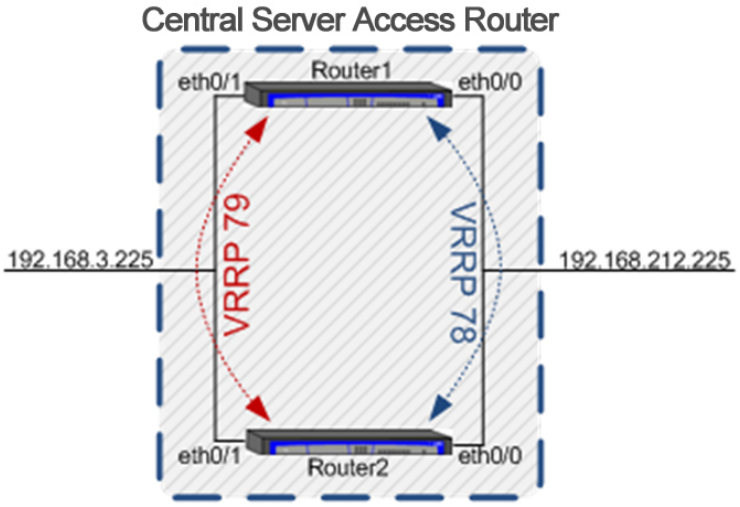
```
Router2 IPSecFT config>enable
Router2 IPSecFT config>show menu
; Showing Menu Configuration for access-level 15 ...
           enable
Router2 IPSecFT config>
```

Now the protocol can establish the relevant sessions between Router1 and Router2 and maintain a database of the sessions that IPsec has set up in each device.

2.4.7.2.3 Configuring VRRP

You need to configure VRRP between the devices that make up the access router to the main server in such a way that they appear to be one router to the outside.

The task of the VRRP protocol is twofold: on the one hand, it decides which device, at any given moment, will have the IP address used to set up the IPsec sessions (VRRP with VRID 79); on the other, it decides which device will communicate with the central server (VRRP with VRID 78). So, for this example, we need to configure two VRRPs on each device, one for each interface.



More information on configuring VRRP can be found in *bintec Dm759-I VRRP Protocol*. Below, we explain certain specific requirements for its use with fault tolerant IPsec recovery:

standby-delay

The *standby-delay* parameter relates to the time it takes a device to assume control of a virtual IP address when it is ready to do so. This parameter is important because a device cannot assume control of a virtual IP address until after the IPsecFT session is established. The reason for this is that VRRP tells IPsecFT when to assume control of the IPsec session, and IPsecFT will not be ready to do so until the session is established.

For the purposes of this example, the *standby-delay* parameter is set to its default value of 10 seconds.

reload-delay

The *reload-delay* parameter governs how long the protocol will wait to initiate once the device has booted up. While the value of this parameter must be different on each of the two routers, it needs to be the same on the interfaces of a single router. This is because we don't want the kind of situation whereby a device cannot simultaneously assume control of both virtual IP addresses when it boots because another device is already doing the same thing.

For this example, the value of the *reload-delay* parameter is 40 seconds for the two interfaces in Router2, and 30 seconds (default value) for the two interfaces in Router1.

Preemption Mode

We recommend disabling the preempt option to minimize changes of ownership of the virtual IP address and thus reduce the number of times the IPsec session changes from one device to another.

```
ip vrrp XX no-preempt
```

VRRP priorities

The option of setting the VRRP priority level to a higher level is another thing to consider here. The reason for this is that VRRP reaction time (which can reach one second) depends on the priority level selected (the higher the priority, the shorter the response time). We recommend using values above 250.

Priority cost in tracking

Finally, we need to remember that both virtual IP addresses must move from one device to another simultaneously. This is achieved by tracking the interfaces on one device to find out when one of them stops working and, if so, reject any virtual IP addresses controlled by the failed interface. This is accomplished by applying the tracking with a priority cost equal to the VRRP priority, i.e., the priority minus the priority cost is 0.

Taking these considerations into account, we can now configure two VRRPs on each device.

In line with the above diagram, we are going to configure each device with a VRID 78 VRRP and a VRID 79 VRRP. VRID 78 will be configured on the ethernet0/0 interfaces on the devices and will control IP address 192.168.212.225, while VRID 79 will be configured on the ehternet0/1 interfaces and will control IP address 192.168.3.225.

The priority level selected for Router1 is 254, and for Router2 is 253.

Both device interfaces are configured to track other interfaces within the same device. When one of the tracked interfaces fails, the tracking interface subtracts an amount from its priority level that is equal to its priority, i.e., it becomes 0.

```
   network ethernet0/0
[…]
      ip vrrp VRID1 priority PRIO

      ip vrrp VRID1 track interface ethernet0/1 prio-cost PRIO
;
   exit
;
   network ethernet0/1
[…]
      ip vrrp VRID2 priority PRIO

      ip vrrp VRDI2 track interface ethernet0/0 prio-cost PRIO
```

The interval between VRRP advertisements for both devices is 100 milliseconds.

```
ip vrrp XX advertise-interval 100 msec
```

The VRRP configuration for each device is as follows:

VRRP in Router1

```
Router1 Config>show config
[…]
   network ethernet0/0
; -- Ethernet Interface User Configuration --
      ip address 192.168.212.219 255.255.254.0
;
      ip vrrp 78 ip 192.168.212.225
      ip vrrp 78 advertise-interval 100 msec
      ip vrrp 78 no-preempt
      ip vrrp 78 accept-vip-packets
      ip vrrp 78 priority 254
      ip vrrp 78 track interface ethernet0/1 prio-cost 254
;
   exit
;
;
```

```
   network ethernet0/1
; -- Ethernet Interface User Configuration --
      ip address 192.168.3.219 255.255.255.0
;
      ip vrrp 79 ip 192.168.3.225
      ip vrrp 79 advertise-interval 100 msec
      ip vrrp 79 no-preempt
      ip vrrp 79 accept-vip-packets
      ip vrrp 79 priority 254
      ip vrrp 79 track interface ethernet0/0 prio-cost 254
;
   exit
[…]
Router1 Config>
```

VRRP in Router2

```
Router2 Config>show config
[…]
   network ethernet0/0
; -- Ethernet Interface User Configuration --
      ip address 192.168.212.218 255.255.254.0
;
      ip vrrp 78 ip 192.168.212.225
      ip vrrp 78 advertise-interval 100 msec
      ip vrrp 78 no-preempt
      ip vrrp 78 accept-vip-packets
      ip vrrp 78 priority 253
      ip vrrp 78 reload-delay 40s
      ip vrrp 78 track interface ethernet0/1 prio-cost 253
;
   exit
;
;
   network ethernet0/1
; -- Ethernet Interface User Configuration --
      ip address 192.168.3.218 255.255.255.0
;
      ip vrrp 79 ip 192.168.3.225
      ip vrrp 79 advertise-interval 100 msec
      ip vrrp 79 no-preempt
      ip vrrp 79 accept-vip-packets
      ip vrrp 79 priority 253
      ip vrrp 79 reload-delay 40s
      ip vrrp 79 track interface ethernet0/0 prio-cost 253
;
   exit
[…]
Router2 Config>
```

2.4.7.2.4 Full configuration

Now that we have expanded on each section of the configuration, we are able to present the entire configuration for Router1 and Router 2:

Router1 configuration

```
Router1 Config>show config
; Showing Menu and Submenus Configuration for access-level 15 ...

   log-command-errors
   no configuration
   set data-link sync serial0/0
   set data-link sync serial0/1
   set hostname Router1
   feature access-lists
; -- Access Lists user configuration --
      access-list 100
         entry 1 default
```

```
        entry 1 permit
        entry 1 source address 192.168.212.0 255.255.254.0
        entry 1 destination address 10.10.2.0 255.255.255.0
;
      exit
;
   exit
;
;
   network ethernet0/0
; -- Ethernet Interface User Configuration --
      ip address 192.168.212.219 255.255.254.0
;
      ip vrrp 78 ip 192.168.212.225
      ip vrrp 78 advertise-interval 100 msec
      ip vrrp 78 no-preempt
      ip vrrp 78 accept-vip-packets
      ip vrrp 78 priority 254
      ip vrrp 78 track interface ethernet0/1 prio-cost 254
;
   exit
;
;
   network ethernet0/1
; -- Ethernet Interface User Configuration --
      ip address 192.168.3.219 255.255.255.0
;
      ip vrrp 79 ip 192.168.3.225
      ip vrrp 79 advertise-interval 100 msec
      ip vrrp 79 no-preempt
      ip vrrp 79 accept-vip-packets
      ip vrrp 79 priority 254
      ip vrrp 79 track interface ethernet0/0 prio-cost 254
;
   exit
;
;
   protocol ip
; -- Internet protocol user configuration --
      route 10.10.2.0 255.255.255.0 192.168.3.100
;
      ipsec
; -- IPSec user configuration --
         enable
         assign-access-list 100
;
         template 1 isakmp tdes sha1
         template 1 ike natt-version rfc
         template 1 keepalive dpd
         template 1 send-original-pkt
;
         template 2 dynamic esp tdes md5
         template 2 source-address 192.168.3.225
         template 2 fault-tolerant
;
         map-template 100 2
         key preshared ip 0.0.0.0 ciphered 0x12D942B46B48645B
         advanced dpd always-send
         fault-tolerant
; -- Fault tolerant IPSec recovery user configuration --
            mode master
            slave-address 192.168.212.218
            source-address 192.168.212.219
            enable
         exit
;
      exit
```

```
;
   exit
;
;
   dump-command-errors
   end
Router1 Config>
```

Router2 configuration

```
Router2 Config>show config
; Showing Menu and Submenus Configuration for access-level 15 ...

   log-command-errors
   no configuration
   set data-link sync serial0/0
   set data-link sync serial0/1
   set hostname Router2
   feature access-lists
; -- Access Lists user configuration --
      access-list 100
         entry 1 default
         entry 1 permit
         entry 1 source address 192.168.212.0 255.255.254.0
         entry 1 destination address 10.10.2.0 255.255.255.0
;
      exit
;
   exit
;
;
   network ethernet0/0
; -- Ethernet Interface User Configuration --
      ip address 192.168.212.218 255.255.254.0
;
      ip vrrp 78 ip 192.168.212.225
      ip vrrp 78 advertise-interval 100 msec
      ip vrrp 78 no-preempt
      ip vrrp 78 accept-vip-packets
      ip vrrp 78 priority 253
      ip vrrp 78 reload-delay 40s
      ip vrrp 78 track interface ethernet0/1 prio-cost 253
;
   exit
;
;
   network ethernet0/1
; -- Ethernet Interface User Configuration --
      ip address 192.168.3.218 255.255.255.0
;
      ip vrrp 79 ip 192.168.3.225
      ip vrrp 79 advertise-interval 100 msec
      ip vrrp 79 no-preempt
      ip vrrp 79 accept-vip-packets
      ip vrrp 79 priority 253
      ip vrrp 79 reload-delay 40s
      ip vrrp 79 track interface ethernet0/0 prio-cost 253
;
   exit
;
;
   protocol ip
; -- Internet protocol user configuration --
      route 10.10.2.0 255.255.255.0 192.168.3.100
;
      ipsec
; -- IPSec user configuration --
         enable
```

```
         assign-access-list 100
;
         template 1 isakmp tdes sha1
         template 1 ike natt-version rfc
         template 1 keepalive dpd
         template 1 send-original-pkt
;
         template 2 dynamic esp tdes md5
         template 2 source-address 192.168.3.225
         template 2 fault-tolerant
;
         map-template 100 2
         key preshared ip 0.0.0.0 ciphered 0x12D942B46B48645B
         advanced dpd always-send
         fault-tolerant
; -- Fault tolerant IPSec recovery user configuration --
            enable
         exit
;
      exit
;
   exit
;
;
   dump-command-errors
   end
Router2 Config>
```

2.4.8 Example 8: IPsec tunnel protection

This scenario shows how to configure an IP tunnel with IPsec protection.

We have two routers and two IPsec tunnels set up between them. The first uses IKEv1 as the negotiation protocol, and the second uses IKEv2.

The first IPsec tunnel protects all traffic sent and received via the tnip1 IP tunnel interface, while the second tunnel protects traffic in the tnip2 IP tunnel.

2.4.8.1 Router1 Configuration

```
Router1 Config$show configuration
; Showing Menu and Submenus Configuration for access-level 15 ...

   log-command-errors
   no configuration
   add device tnip 1
   add device tnip 2
   add device loopback 1
   set data-link sync serial0/0
   set data-link sync serial0/1
   set hostname Router1
   set inactivity-timer disabled
;
   network ethernet0/0
; -- Ethernet Interface User Configuration --
      ip address 192.168.0.1 255.255.255.0
      ip address 10.10.10.203 255.255.255.0 secondary
;
   exit
;
;
;
   network tnip1
; -- IP Tunnel Net Configuration --
      ip address 1.1.1.203 255.255.255.0
;
      mode ipsec ip
      source 192.168.0.1
      destination 192.168.0.2
      protection-ipsec
   exit
;
;
   network tnip2
; -- IP Tunnel Net Configuration --
      ip address 2.2.2.203 255.255.255.0
;
      mode ipsec ip
      source 10.10.10.203
      destination 10.10.10.205
      protection-ipsec
   exit
;
;
   network loopback1
; -- Loopback interface configuration --
      ip address 2.3.4.5 255.255.255.0
;
   exit
;
   protocol ip
; -- Internet protocol user configuration --
      ipsec
; -- IPSec user configuration --
         enable
;
         template 1 isakmp des md5
         template 1 ike natt-version rfc
         template 1 keepalive dpd
;
         template 2 dynamic esp des md5
         template 2 tunnel-protection tnip2
;
         template 3 ikev2 encryption tdes
```

```
         template 3 ikev2 authentication md5
         template 3 ike natt-version rfc
         template 3 ike method local preshared
         template 3 ike method remote preshared
         template 3 keepalive dpd
;
         template 4 dynamic esp des md5
         template 4 negotiation-protocol ikev2
         template 4 tunnel-protection tnip1
;
         key preshared ip 0.0.0.0 ciphered 0xE51A70141339BFED
         advanced dpd no always-send
      exit
;
   exit
;
   protocol rip
; -- RIP protocol user configuration --
      enable
   exit
;
;
   dump-command-errors
   end
Router1 Config$
```

2.4.8.2 Router2 Configuration

```
Router2 Config$show configuration
; Showing Menu and Submenus Configuration for access-level 15 ...

   log-command-errors
   no configuration
   add device tnip 1
   add device tnip 2
   add device loopback 1
   set data-link sync serial0/0
   set data-link sync serial0/1
   set hostname Router1
   set inactivity-timer disabled
;
   network ethernet0/0
; -- Ethernet Interface User Configuration --
      ip address 192.168.0.2 255.255.255.0
      ip address 10.10.10.205 255.255.255.0 secondary
;
   exit
;
;
;
   network tnip1
; -- IP Tunnel Net Configuration --
      ip address 1.1.1.205 255.255.255.0
;
      mode ipsec ip
      source 192.168.0.2
      destination 192.168.0.1
      protection-ipsec
   exit
;
;
   network tnip2
; -- IP Tunnel Net Configuration --
      ip address 2.2.2.205 255.255.255.0
;
      mode ipsec ip
      source 10.10.10.205
      destination 10.10.10.203
      protection-ipsec
   exit
;
;
   network loopback1
; -- Loopback interface configuration --
      ip address 6.7.8.9 255.255.255.0
;
   exit
;
   protocol ip
; -- Internet protocol user configuration --
      ipsec
; -- IPSec user configuration --
         enable
;
         template 1 isakmp des md5
         template 1 ike natt-version rfc
         template 1 keepalive dpd
;
         template 2 dynamic esp des md5
         template 2 tunnel-protection tnip2
;
         template 3 ikev2 encryption tdes
```

```
         template 3 ikev2 authentication md5
         template 3 ike natt-version rfc
         template 3 ike method local preshared
         template 3 ike method remote preshared
         template 3 keepalive dpd
;
         template 4 dynamic esp des md5
         template 4 negotiation-protocol ikev2
         template 4 tunnel-protection tnip1
;
         key preshared ip 0.0.0.0 ciphered 0xE51A70141339BFED
         advanced dpd no always-send
      exit
;
   exit
;
   protocol rip
; -- RIP protocol user configuration --
      enable
   exit
;
;
   dump-command-errors
   end
Router2 Config$
```

2.4.9 Example 9: IKEv2 IPsec tunnel protection and Route Injection

This scenario shows how to configure an IP tunnel with IPsec protection using IKEv2 as the negotiation protocol and injecting routes via the IKEv2 configuration payload.

In this case, we have two routers with one IPsec tunnel set up between them. Said IPsec tunnel protects all traffic sent and received via the tnip1 IP tunnel interface, which is configured with GRE encapsulation (tunnel running in GRE/IP mode).

By using GRE as the data encapsulation method, IPv4 and IPv6 traffic can be transmitted through the tunnel. In addition, devices send IPv4 and IPv6 routes to the remote end via the IKEv2 configuration payload.

2.4.9.1 Router1 Configuration

```
Router1 Config$show configuration
; Showing Menu and Submenus Configuration for access-level 15 ...

   log-command-errors
   no configuration
   add device tnip 1
   add device loopback 1
   add device loopback 2
   set data-link sync serial0/0
   set data-link sync serial0/1
   set hostname Router1
   set inactivity-timer disabled
   feature ipv6-access-list
; -- IPv6 Access Lists user configuration --
      access-list list1
         entry 1 deny
         entry 1 source address 3333::3333/64
;
      exit
;
   exit
;
   feature access-lists
; -- Access Lists user configuration --
      access-list 1
         entry 1 default
         entry 1 permit
         entry 1 source address 33.33.33.33 255.255.255.255
;
      exit
;
   exit
;
;
   network ethernet0/0
; -- Ethernet Interface User Configuration --
      ip address 192.168.0.1 255.255.255.0
;
   exit
;
;
;
   network tnip1
; -- IP Tunnel Net Configuration --
      ip address unnumbered loopback1
;
      ipv6 address 1111::1111/64
      mode gre ip
      source ethernet0/0
      destination 192.168.0.2
      protection-ipsec
   exit
;
;
   network loopback1
; -- Loopback interface configuration --
      ip address 2.3.4.5 255.255.255.0
;
   exit
;
   network loopback2
; -- Loopback interface configuration --
      ip address 33.33.33.33 255.255.255.255
;
```

```
      ipv6 address 3333::3333/64
   exit
;
   protocol ip
; -- Internet protocol user configuration --
      ipsec
; -- IPSec user configuration --
         enable
;
         template 1 ikev2
         template 1 ike natt-version rfc
         template 1 ike method local preshared
         template 1 ike method remote preshared
         template 1 config route-accept metric 30
         template 1 config route-set inteface
         template 1 config route-set 1
         template 1 config route-set ipv6-access-list list1
         template 1 keepalive dpd
;
         template 2 dynamic esp aes128 sha1
         template 2 negotiation-protocol ikev2
         template 2 tunnel-protection tnip1
;
         key preshared ip 0.0.0.0 ciphered 0xE51A70141339BFED
         advanced dpd no always-send
      exit
;
   exit
;
   dump-command-errors
   end
Router1 Config$
```

2.4.9.2 Router2 Configuration

```
Router2 Config$show configuration
; Showing Menu and Submenus Configuration for access-level 15 ...

   log-command-errors
   no configuration
   add device tnip 1
   add device loopback 1
   add device loopback 2
   set data-link sync serial0/0
   set data-link sync serial0/1
   set hostname Router1
   set inactivity-timer disabled
   feature ipv6-access-list
; -- IPv6 Access Lists user configuration --
      access-list list1
         entry 1 deny
         entry 1 source address 4444::4444/64
;
      exit
;
   exit
;
   feature access-lists
; -- Access Lists user configuration --
      access-list 1
         entry 1 default
         entry 1 permit
         entry 1 source address 44.44.44.44 255.255.255.255
      exit
;
   exit
;
;
   network ethernet0/0
; -- Ethernet Interface User Configuration --
      ip address 192.168.0.2 255.255.255.0
;
   exit
;
;
;
   network tnip1
; -- IP Tunnel Net Configuration --
      ip address unnumbered loopback1
;
      ipv6 address 2222::2222/64
      mode gre ip
      source ethernet0/0
      destination 192.168.0.1
      protection-ipsec
   exit
;
;
;
   network loopback1
; -- Loopback interface configuration --
      ip address 6.7.8.9 255.255.255.0
;
   exit
;
   network loopback2
; -- Loopback interface configuration --
      ip address 44.44.44.44 255.255.255.255
;
```

```
      ipv6 address 4444::4444/64
   exit
;
;
   protocol ip
; -- Internet protocol user configuration --
      ipsec
; -- IPSec user configuration --
         enable
;
         template 1 ikev2
         template 1 ike natt-version rfc
         template 1 ike method local preshared
         template 1 ike method remote preshared
         template 1 config route-accept metric 30
         template 1 config route-set inteface
         template 1 config route-set 1
         template 1 config route-set ipv6-access-list list1
         template 1 keepalive dpd
;
         template 2 dynamic esp aes128 sha1
         template 2 negotiation-protocol ikev2
         template 2 tunnel-protection tnip1
;
         key preshared ip 0.0.0.0 ciphered 0xE51A70141339BFED
         advanced dpd no always-send
      exit
;
   exit
;
   dump-command-errors
   end
Router2 Config$
```

2.5 Certificates

RSA asymmetric keys must be used when using RSA-based authentication. These keys are usually used within higher-level encapsulations called *certificates*. Our routers allow RSA-based authentication, but you will need to install certain certificate management tools.

Note

The device only uses certificates that are valid, correctly signed, and that have not been revoked.

If these conditions are not met, the device still loads certificates and displays their properties but it doesn't use them.

Note

A certificate must be valid in order to be used, so please ensure that the device time is properly synchronized and that the time zone parameters and daylight saving times are configured correctly.

In this section, we will explain how to work with certificates, i.e., how they are downloaded, assigned to templates, created, etc.

2.5.1 CERT menu

The CERT menu is located in the IPsec menu and has the following commands:

```
IPSec config>cert

-- Cert user configuration --
CERTIFICATES config>?
  certificate    Certificate operations
```

```
  crl            Certificate Revocation List menu
  csr            Certificate Signed Request menu
  file           Certificate file block
  list           List certificates
  scep           Simple Certificate Enrollment menu
  exit           Exits Certificate configuration menu
CERTIFICATES config>
```

The **certificate** command from the CERT menu has the following options:

Command	Operation
LOAD	Loads a CERTIFICATE from a disk to RAM memory.
DISK_DELETE	Deletes a CERTIFICATE from a disk.
CONFIG_DELETE	Deletes a CERTIFICATE from the configuration.
PRINT	Displays the content of a CERTIFICATE on screen.
BASE64	Loads a CERTIFICATE from the console in base64 format.
PKCS12	PKCS12 managing. This option will be explained in *PKCS #12. Personal Information Exchange* on page 150
NO	Disables or deletes an option.

“CERTIFICATE [CertFile] LOAD”

Allows you to load a certificate from a disk into the device's RAM memory. A certificate must be downloaded into the RAM using this command before you can use it to perform an operation.

Example:

```
CERTIFICATES config>certificate router.cer load
```

“CERTIFICATE [CertFile] DISK_DELETE”

Deletes a certificate from a disk. The certificates can be loaded from a file stored on a disk or from the router configuration using the *FILE* command.

Example:

```
CERTIFICATES config>certificate router.cer disk_delete
```

“CERTIFICATE [CertFile] CONFIG_DELETE”

Allows you to delete a certificate from the configuration. The certificates can be loaded from a file stored on a disk or from the router configuration using the *FILE* command.

Example:

```
CERTIFICATES config>certificate router.cer config_delete
```

“CERTIFICATE [CertFile] PRINT”

Allows you to print the contents of a previously loaded certificate.

Example:

```
CERTIFICATES config>certificate router.cer print
Version                   : V3
Serial Number             : 547E D185 0000 0000 1E6E
Algorithm Identifier      : SHA1 With RSA
Issuer:
  CN (Common Name        ): SECTESTCA1
  OU (Organizational Unit): Microsoft, Interoperability Testing Only
  O  (Organization Name  ): Microsoft, Interoperability Testing Only
  L  (Locality           ): Redmond
  S  (State or Province  ): WA
  C  (Country Name       ): US
  E  (Email              ): testca@microsoft.com
Valid From                : Wed Jul 25 09:21:24 2001
Valid To                  : Thu Jul 25 09:31:24 2002
Subject:
  E  (Email              ): jiglesias@bintec.de
  CN (Common Name        ): router.bintec.de
  OU (Organizational Unit): ImasD
```

```
  O  (Organization Name  ): Bintec
  L  (Locality           ): Tres Cantos
  S  (State or Province  ): Madrid
  C  (Country Name       ): sp
Public Key               :
  Algorithm Identifier    : RSA
  Modulus Length          : 512 Bits.
  Modulus                 :
    E1CF D175 90EE 43BC 4BC5 D215 695A 74CC D1E8 F301 4F09 2093 7B12 84C0
    2C07 DE4B E458 9D48 43CB 4F14 A075 0D09 FB57 71DB 4FC6 8FDF 1FEF AA6D
    13BB 96FB 88FA 1343
  Exponent                :
    01 00 01
Signature                 :
  Signature Algorithm     : SHA1 With RSA
  Signature Data Info     : 2048 Bits.
  Signature Data          :
    3C10 94F3 CE87 0040 C3D0 A59F 1F0E 84DC E21F CCFD CA7A 2A32 651B 3D27 F9D0
    F87A 6993 E22C 28F5 7954 ED49 1E90 A52C 8098 F686 5E51 18DA D713 D65E 81BB
    267A 1D70 957D FB2F C841 E155 AD3C 3B38 6796 FA62 F6EF 8D76 DEDF 09B2 52C3
    3496 AD4B BF06 1415 3111 DEDD B2BE 9C68 5584 0A3B BF41 90B3 05C4 5CA1 E079
    AADA 43B1 F48D 9DEE 9793 907E 262D 2CC5 325C F3D1 892C 54E7 4736 06A3 4883
    A239 B68D 5477 13A8 BDE0 D7F4 18C1 FD94 3116 48FC C701 BA86 D932 A5C8 C28C
    5FE0 D8CF BE39 CF77 5CCC A104 0189 FF0B 5598 DBB1 2EB5 6269 9683 31DF 19BB
    DDEB 8BC0 FFDA 4587 13E4 42FF 7AF1 BD63 ACE4 D469 37B7 03FA 78DD 4535 49FB
    36AA 4525 F6EF 33A8 F5DB 3934 5079 A536
```

“CERTIFICATE [CertFile] BASE64”

Allows you to import a certificate in base64 format. Once you have executed the command without any errors, the certificate will be saved in the configuration and displayed after the *FILE* command.

Example:

```
CERTIFICATES config>cert wiscon base64

Introduce the Certificate (Base 64 format)
Enter <cr> to escape
-----BEGIN CERTIFICATE-----
MIIC6zCCAlSgAwIBAgICAlQwDQYJKoZIhvcNAQEEBQAwgakxCzAJBgNVBAYTAlVT
MRIwEAYDVQQIEwlXaXNjb25zaW4xEDAOBgNVBAcTB01hZGlzb24xIDAeBgNVBAoT
F1VuaXZlcnNpdHkgb2YgV2lzY29uc2luMSswKQYDVQQLEyJEaXZpc2lvbiBvZiBJ
bmZvcm1hdGlvbiBUZWNobm9sb2d5MSUwIwYDVQQDExxIRVBLSSBNYXN0ZXIgQ0Eg
LS0gMjAwMjA3MDFBMB4XDTAyMDYzMDIyMTYzOVoXDTI5MTExNjIyMTYzOVowgakx
CzAJBgNVBAYTAlVTMRIwEAYDVQQIEwlXaXNjb25zaW4xEDAOBgNVBAcTB01hZGlz
b24xIDAeBgNVBAoTF1VuaXZlcnNpdHkgb2YgV2lzY29uc2luMSswKQYDVQQLEyJE
aXZpc2lvbiBvZiBJbmZvcm1hdGlvbiBUZWNobm9sb2d5MSUwIwYDVQQDExxIRVBL
SSBNYXN0ZXIgQ0EgLS0gMjAwMjA3MDFBMIGfMA0GCSqGSIb3DQEBAQUAA4GNADCB
iQKBgQDJ3FDZym9Ja94DP7TUZXf3Vu3CZwqZzYThgjUT2eBJBYVALISSJ+RjJ2j2
CYpq3wesSgWHqfrpPnTgTBvn5ZZF9diX6ipAmC0H75nySDY8B5AN1RbmPsAZ51F9
7Eo+6JZ59BFYgowGXyQpMfhBykBSySnvnOX5ygTCz20LwKkErQIDAQABoyAwHjAP
BgNVHRMBAf8EBTADAQH/MAsGA1UdDwQEAwIBpjANBgkqhkiG9w0BAQQFAAOBgQB1
8ZXB+KeXbDVkz+b2xVXYmJiWrp73IOvi3DuIuX1n88tbIH0ts7dJLEqr+c0owgtu
QBqLb9DfPG2GkJ1uOK75wPY6XWusCKDJKMVY/N4ec9ew55MnDlFFvl4C+LkiS2YS
Ysrh7fFJKKp7Pkc1fxsusK+MBXjVZtq0baXsU637qw==
-----END CERTIFICATE-----
```

2.5.2 KEY RSA command

Available from the IPsec menu, this command allows you to work with the RSA keys generated in the router.

```
IPSec config>key rsa ?
  generate            Generates an RSA key
  ca-change           Changes an RSA certificate
  extended-ca-name    Set an extended Certificate Authority name
IPSec config>
```

Command	Operation
GENERATE	Generates a random RSA key pair.

CA-CHANGE	Changes the certificate authority associated with the generated RSA key.
EXTENDED-CA-NAME	Sets an extended certificate authority name.

“KEY RSA GENERATE [CA NAME][SIZE(512/1024/2048/4096)]”

Allows you to generate a random RSA key and assign it to a certificate authority (CA). In other words, it lets you generate a public/private key pair that will be stored to the device's disk when the configuration is saved.

After generating the key pair, the device suggests generating a *Certificate Signing Request* (CSR), telling the user to go to the CSR menu from the CERT menu and execute the *MAKE [RSA KEY ID]* command.

Example:

```
IPSec config>key rsa generate caname 512
RSA Key Generation.
Please, wait for a few seconds.
 RSA Key Generation done.
Checking..OK
Key Generation Process Finished. RSA Key ID: 1
Do not forget to save RSA keys.

It's a good moment to make the Certificate Signing Request (CSR) associated with this RSA Key...
If you want to do it, go to the “CERT” menu and then to the “CSR” menu and execute the command “MAKE 1”.

IPSec config>
```

Important

While the generated RSA keys are stored in the device configuration, they are not displayed with the *SHOW CONFIG* command for security reasons. Therefore, if you want to modify text that you have copied after running the *SHOW CONFIG* command and then paste it into another device, the RSA keys will not be copied along with the rest of the configuration.

Command history:

Release	Modification
10.08.34.05.08	As of version 10.08.34.05.08, a 4096-bit key size is supported.
10.08.43	As of version 10.08.43, a 4096-bit key size is supported.
10.09.23	As of version 10.09.23, a 4096-bit key size is supported.
11.00.00.02.06	As of version 11.00.00.02.06, a 4096-bit key size is supported.
11.00.03	As of version 11.00.03, a 4096-bit key size is supported.
11.01.00	As of version 11.01.00, a 4096-bit key size is supported.

“KEY RSA CA-CHANGE”

Allows you to change the certificate authority (CA) associated with a previously generated RSA key.

Example:

```
IPSec config>list key rsa
1 rsakey entries
Id.           Date.         Len         Certificate Authority (CA)           Cert sn.
   1   06/18/03  11:46:16    512         CANAME.CER                            ---
IPSec config>key rsa ca-change 1 newca
Do not forget to save RSA keys changes.
IPSec config>list key rsa
1 rsakey entries
Id.           Date.         Len         Certificate Authority (CA)           Cert sn.
   1   06/18/03  11:46:16    512           NEWCA.CER                           ---
```

“KEY RSA EXTENDED-CA-NAME”

This command is similar to the previous command in that it also allows you to change the certificate authority (CA) associated with a previously generated RSA key. However, it also allows you to extend the length to more than 8 characters.

Example:

```
IPSec config>list key rsa
1 rsakey entries
Id.           Date.         Len         Certificate Authority (CA)           Cert sn.
   1   06/18/03  11:46:16    512         CANAME.CER                            ---
IPSec config>key rsa extended-ca-name 1 canamelong
Do not forget to save RSA keys changes.
IPSec config>list key rsa
1 rsakey entries
Id.           Date.         Len         Certificate Authority (CA)           Cert sn.
   1   06/18/03  11:46:16    512         CANAMELONG.CER                        ---
```

2.5.3 Obtaining certificates through CSR

You can obtain a certificate for one of our devices by generating a Certificate Signing Request (CSR). Ultimately, you want to get two files: the certificate authority (CA) certificate, *caname.cer*, and the router certificate, *router.cer* (if these certificates haven't come from the root CA, you will have to install all the certificates along the path up to the root CA). The steps are as follows:

- Configure the various CSR attribute values, i.e., *subject-name* (**mandatory**), *alternative-name* (optional) and *password* (optional), and select the *signature algorithm* that will be used to generate the certificate. For this configuration, you need to access the CSR setup menu and execute the respective commands. These commands will be described in detail in the section entitled "CSR menu":
- *subject-name ["C=Country, L=Locality, ST=State or Province, O=Organization, OU=Organization Unit, CN=Common Name, E=Email ... "]*. This attribute is mandatory to generate the CSR.
- *alt-name [Alternative name]*. This attribute is optional.
- *password ciphered/plain [Password]* . This attribute is also optional.
- *Signature-algorithm MD5/SHA1/MD2/SHA-256/SHA-384/SHA-512* . MD5 is the default algorithm used.

(1) If you have generated a private key, you should also generate a CSR and associate it with the private key. To do this, run the *MAKE* command followed by the private key ID from the CSR setup menu (*MAKE [RSA Key ID]*). If you haven't already generated a private key, you should generate one using the *KEY RSA GENERATE* command. The private key will be associated with a CA using a file name that corresponds to the certificate installed in the CA's device, *caname.cer*. You can perform this operation even if the CA certificate is not yet available.

(2) After generating the CSR, you can save it in a file. You can obtain the file later via FTP or by running the *PRINT* command in the console. CSRs are usually encoded in base64 format.

(3) Submit the CSR to the CA in order to get the *router.cer* certificate. At this point, the CA usually sends its own *caname.cer* CA certificate as well.

(4) Install the certificates in the device, sending them via FTP and running the *QUOTE SITE SAVEBUFFER* command, or using the *CERTIFICATE [CertFile] BASE64* command.

(5) Create a template that uses the RSA method, TEMPLATE 1 IKE METHOD RSA.

(6) Run the *TEMPLATE 1 IKE CA* **caname** command to assign the CA certificate to the template you are using.

(7) Finally, you must save the settings.

Thus, the components are assigned as follows:

- **(Private key, CA)** = Assigned using the CA name.
- **(Private key, CSR)** = Assigned using the private key identifier.
- **(Private key, device certificate)** = Assigned using the CA and the certificate serial number. The CA must be assigned to a template and the certificate must have been downloaded.

Note

Verisign does not accept certain characters in the CSR fields. One such character is the @ symbol, which means you cannot include an email address. You will get error 105 from Verisign if you try to insert an invalid character. This field should be left blank if you are going to send your CSR to Verisign.

The *LIST TEMPLATE ALL* command will show how it all went:

```
IPSec config>list template all
TEMPLATES
1 isakmp 3DES MD5  DES=1.1.1.1
   LifeTime:1h0m0s
   IKE MAIN
   RSA SIGNATURE
```

```
    CA      : SECTEST.CER. Expired.
    CRL     : disabled
     USER    : ROUTER.CER. Signature ok. Expired. Without Private Key.
  fqdn ID TYPE
  OAKLEY GROUP 1
  DPD enabled
```

2.5.4 CSR menu

The Certificate Signing Request (CSR) setup menu is accessed through the *CSR* command from the CERT menu. It contains the commands for generating CSRs and configuring CSR attributes:

Command	Function
ALT-NAME	Configures the CSR alternative-name attribute.
CLEAN	Deletes a CSR from RAM.
DELETE	Deletes a CSR file stored on disk.
LIST	Displays the CSR files, stored on disk, on screen.
LOAD	Loads a CSR file from disk to RAM.
MAKE	Generates a CSR.
NO	Deletes an option or restores its default value.
PASSWORD	Configures the CSR password attribute.
PRINT	Displays the contents of a CSR on screen.
SIGNATURE-ALGORITHM	Configures the signature-algorithm used in the CSR.
SUBJECT-NAME	Configures the CSR subject-name.
EXIT	Exits the CSR setup menu.

“ALT-NAME [Alternative-name]”

Allows you to configure the *alternative-name* attribute that will be included in the CSR that you generate later on.

Use the *NO ALT-NAME* command to delete the configured *alternative-name*.

Example:

```
CSR config>alt-name ?
  <1..128 chars>    Alternative-name text
CSR config>alt-name bintec.de
CSR config>
```

“CLEAN”

Deletes the stored CSR from RAM.

Example:

```
CSR config>clean
CSR cleaned OK
CSR config>
```

“DELETE [file name]”

Deletes a CSR file stored on disk.

Example:

```
CSR config>delete ?
  <word>    File name
CSR config>delete prueba
CSR successfully deleted from disk
CSR config>
```

“LIST”

Displays a list of the CSR files stored on disk.

Example:

```
CSR config>list
```

```
 A:                    PRUEBA.CSR        494   06/25/09   10:38   Flash
 A:                   PRUEBA2.CSR        494   06/25/09   10:38   Flash
CSR config>
```

“LOAD [file name]”

Loads a CSR file stored on disk to RAM.

Example:

```
CSR config>load ?
  <word>    File name
CSR config>load prueba
 CSR loaded
CSR config>
```

“MAKE [RSA Key ID] [yes]”

Associates a CSR that we've generated with the private key of the ID entered. As you can see below, the *subject-name* attribute must be set before you can generate the CSR. Once you have generated the CSR, the router displays it on screen and asks whether you want to store it to disk. If option **yes** is entered, this question is omitted and accepted automatically.

Example:

```
CSR config>make ?
  <1..65535>    RSA Key ID for the CSR request
CSR config>make 1
  Certificate Request
==================
-----BEGIN CERTIFICATE REQUEST-----
MIIBdTCCAR8CAQAwgYwxCzAJBgNVBAYTAmVzMQ8wDQYDVQQHEwZtYWRyaWQxDzAN
BgNVBAoTBnRlbGRhdDEOMAwGA1UECxMFaW1hc2QxEDAOBgNVBAsTB3BsYW50YTEx
GDAWBgNVBAMTD3RlbGRhdC5pbWFzZC5lczEfMB0GCSqGSIb3DQEJARMQcHJ1ZWJh
QHRlbGRhdC5lczBcMA0GCSqGSIb3DQEBAQUAA0sAMEgCQQDCCyRNzTmf9rc8OBn8
72T1Gl75lBdAUYK6G5IlQWxVvk2wn0+4wvsEHB2nvUiCrp/dxdhTnmdBZO/Q0nFS
uRzpAgMBAAGgLTArBgkqhkiG9w0BCQ4xHjAcMBoGA1UdEQQTMBGCD3RlbGRhdC5p
bWFzZC5lczANBgkqhkiG9w0BAQQFAANBAEoKwrsADiSfKt5hWPunuBEwxRhFRz7S
Ty4ykAUFaNuneaq7T6XCz77uszsVt4v5QQJ8N7M7wr0dQBIBNVxbwbY=
-----END CERTIFICATE REQUEST-----
Save in file(Yes/No)? n
CSR config>
```

Command history:

Release	Modification
11.01.06	The "[yes]" option was added as of version 11.01.06

“PASSWORD CIPHERED/PLAIN [Password]”

Configures the **password** attribute that will be included in the CSR that you generate later on. The password can be entered in encrypted format by running *CIPHERED* (subcommand) or in clear using the **PLAIN** subcommand.

Use the *NO PASSWORD* command to delete the configured password.

Example:

```
CSR config>password ?
  ciphered    Ciphered password
  plain       Plain password
CSR config>password pla
CSR config>password plain ?
  <4..95 chars>    Text
CSR config>password plain mypassword
CSR config>
```

“PRINT ASN.1/BASE64/READABLE”

Displays the contents of a CSR stored in RAM, on screen. It can be displayed in various formats: ASN1, BASE64 or READABLE.

Syntax:

```
CSR config>print ?
  asn.1       ASN.1 format
  base64      BASE64 format
  readable    Readable format
CSR config>
```

Example 1:

```
CSR config>print asn.1
Certificate Request
===================
30 82 01 75 30 82 01 1F 02 01 00 30 81 8C 31 0B 30 09 06 03
55 04 06 13 02 65 73 31 0F 30 0D 06 03 55 04 07 13 06 6D 61
64 72 69 64 31 0F 30 0D 06 03 55 04 0A 13 06 74 65 6C 64 61
74 31 0E 30 0C 06 03 55 04 0B 13 05 69 6D 61 73 64 31 10 30
0E 06 03 55 04 0B 13 07 70 6C 61 6E 74 61 31 31 18 30 16 06
03 55 04 03 13 0F 74 65 6C 64 61 74 2E 69 6D 61 73 64 2E 65
73 31 1F 30 1D 06 09 2A 86 48 86 F7 0D 01 09 01 13 10 70 72
75 65 62 61 40 74 65 6C 64 61 74 2E 65 73 30 5C 30 0D 06 09
2A 86 48 86 F7 0D 01 01 01 05 00 03 4B 00 30 48 02 41 00 C2
0B 24 4D CD 39 9F F6 B7 3C 38 19 FC EF 64 F5 1A 5E F9 94 17
40 51 82 BA 1B 92 25 41 6C 55 BE 4D B0 9F 4F B8 C2 FB 04 1C
1D A7 BD 48 82 AE 9F DD C5 D8 53 9E 67 41 64 EF D0 D2 71 52
B9 1C E9 02 03 01 00 01 A0 2D 30 2B 06 09 2A 86 48 86 F7 0D
01 09 0E 31 1E 30 1C 30 1A 06 03 55 1D 11 04 13 30 11 82 0F
74 65 6C 64 61 74 2E 69 6D 61 73 64 2E 65 73 30 0D 06 09 2A
86 48 86 F7 0D 01 01 04 05 00 03 41 00 4A 0A C2 BB 00 0E 24
9F 2A DE 61 58 FB A7 B8 11 30 C5 18 45 47 3E D2 4F 2E 32 90
05 05 68 DB A7 79 AA BB 4F A5 C2 CF BE EE B3 3B 15 B7 8B F9
41 02 7C 37 B3 3B C2 BD 1D 40 12 01 35 5C 5B C1 B6
CSR config>
```

Example 2:

```
CSR config>print base64
Certificate Request
===================
-----BEGIN CERTIFICATE REQUEST-----
MIIBdTCCAR8CAQAwgYwxCzAJBgNVBAYTAmVzMQ8wDQYDVQQHEwZtYWRyaWQxDzAN
BgNVBAoTBnRlbGRhdDEOMAwGA1UECxMFaW1hc2QxEDAOBgNVBAsTB3BsYW50YTEx
GDAWBgNVBAMTD3RlbGRhdC5pbWFzZC5lczEfMB0GCSqGSIb3DQEJARMQcHJ1ZWJh
QHRlbGRhdC5lczBcMA0GCSqGSIb3DQEBAQUAA0sAMEgCQQDCCyRNzTmf9rc8OBn8
72T1Gl75lBdAUYK6G5IlQWxVvk2wn0+4wvsEHB2nvUiCrp/dxdhTnmdBZO/Q0nFS
uRzpAgMBAAGgLTArBgkqhkiG9w0BCQ4xHjAcMBoGA1UdEQQTMBGCD3RlbGRhdC5p
bWFzZC5lczANBgkqhkiG9w0BAQQFAANBAEoKwrsADiSfKt5hWPunuBEwxRhFRz7S
Ty4ykAUFaNuneaq7T6XCz77uszsVt4v5QQJ8N7M7wr0dQBIBNVxbwbY=
-----END CERTIFICATE REQUEST-----
CSR config>
```

Example 3:

```
CSR config>print readable
Certificate Request
===================
Version                    : V1
Subject:
  E  (Email               ): prueba@bintec.de
  CN (Common Name         ): bintec.randd.de
  OU (Organizational Unit): planta1
  OU (Organizational Unit): imasd
  O  (Organization Name   ): bintec
  L  (Locality            ): madrid
  C  (Country Name        ): es
Public Key                 :
  Algorithm Identifier     : RSA
  Modulus Length           : 512 Bits.
  Modulus                  :
    C20B 244D CD39 9FF6 B73C 3819 FCEF 64F5 1A5E F994 1740 5182 BA1B 9225
    416C 55BE 4DB0 9F4F B8C2 FB04 1C1D A7BD 4882 AE9F DDC5 D853 9E67 4164
```

```
    EFD0 D271 52B9 1CE9
  Exponent                :
    01 00 01
CSR Attributes            :
  1.2.840.113549.1.9.14
    311E301C301A0603551D110413301182 : 1.0.0...U....0..
    0F74656C6461742E696D6173642E6573 : .bintec.randd.de
CSR Signature
  CSR Signature Algorithm : MD5 With RSA
  CSR Signature Data Info : 512 Bits.
  Signature Data          :
    4A0A C2BB 000E 249F 2ADE 6158 FBA7 B811 30C5 1845 473E D24F 2E32 9005 0568
    DBA7 79AA BB4F A5C2 CFBE EEB3 3B15 B78B F941 027C 37B3 3BC2 BD1D 4012 0135
    5C5B C1B6
CSR config>
```

“SIGNATURE-ALGORITHM”

Lets you select the encryption algorithm that will be used for signing the CSRs. The algorithm can be MD5, SHA1, MD2, SHA-256, SHA-384, or SHA-512. The default is MD5.

Use the *NO SIGNATURE-ALGORITHM* command to restore the default value.

Example:

```
CSR config>signature-algorithm ?
  md5       Set Signature Algorithm of CSR to MD5 algorithm
  sha1      Set Signature Algorithm of CSR to SHA1 algorithm
  md2       Set Signature Algorithm of CSR to MD2 algorithm
  sha256    Set Signature Algorithm of CSR to SHA256 algorithm
  sha384    Set Signature Algorithm of CSR to SHA384 algorithm
  sha512    Set Signature Algorithm of CSR to SHA512 algorithm
CSR config>signature-algorithm sha1
CSR config>
```

“SUBJECT-NAME [Subject-name in X500 format]”

Configures the *subject-name* attribute that will be included in the CSR that you generate later on. You must configure this attribute in order to generate the CSR.

The *subject-name* is entered as a character string separated by commas and in X500 format, as shown below:

X500 Format: " **C** = Country**, L** = Locality**, ST** = State or Province**, O** = Organization,

OU = Organization Unit**, CN** = Common Name**, E** = Email ... "

Once configured, the **subject-name** can be modified (i.e., have new fields added to it) by executing the command again and storing it on different lines. This attribute can have a maximum of 20 fields and you can enter multiple fields of the same type.

Use the *NO SUBJECT-NAME* command to delete the configured subject-name.

Example:

```
CSR config>subject-name ?
  <1..250 chars>    Subject-name: " C=Country, L=Locality, ST=State/Province,
                    O=Organization, OU=OrgUnit, CN=CommonName, E=Email,
                    DC=DomainComponent... "
CSR config>subject-name "c = es, l = madrid, o = bintec, ou = imasd, ou = planta1"
CSR config>subject-name "cn = bintec.randd.de, e = prueba@bintec.de"
CSR config>
```

Note

1. Be sure to enter the backslash character (\) before entering the equals sign (=), backslash (\) or comma (,) in any of the fields.

2. Inverted commas (“) are not permitted in any of the fields.

“EXIT”

Lets you return to the CERT menu from the CSR menu.

Example:

```
CSR config>exit
CERTIFICATES config>
```

2.5.5 Obtaining certificates through SCEP

You can obtain a certificate for one of our devices through the *Simple Certificate Enrollment Protocol*, SCEP. Ultimately, you want to get two files: the CA certificate, *caname.cer*, and the router certificate, *router.cer* (if these certificates haven't come from the root CA, you will have to install all the certificates along the path up to the root CA). This method of obtaining certificates is an alternative to the method given in earlier sections of obtaining certificates through CSR. Please read the section on *Obtaining certificates through CSR* on page 132 before continuing.

The SCEP protocol allows you to obtain the certificates by connecting to a server (usually the CA server). The device establishes an HTTP connection to the server and the information is exchanged over this connection.

The following processes form the basis of the protocol:

(1) Installing the CA certificate: *Install*

(2) Installing the user certificate: *Enroll*

Install

The first thing you have to do is to install the CA certificate on the device. There are several ways to do this:

- Through the configuration: use the *FILE* command from the CERT menu to enter the commands for defining a certificate file:

Example:

```
CERTIFICATES config>file new BINTECCA.CER
CERTIFICATES config>file add 0x308202AF30820218A003020102020101300D06092A864886F70D010104050030
CERTIFICATES config>file add 0x6B310B3009060355040613025553310B300906035504081302 4E433110300E06
CERTIFICATES config>file add …
CERTIFICATES config>file end 0xC1809E37BB050F7D27DB2C2ACC8AD4
```

- Loading a file in base64 format using the *CERTIFICATE [CertFile] BASE64* command from the CERT menu.
- Loading the file through ftp and executing the *CERTIFICATE [CertFile] BASE64 LOAD* command from the CERT menu.
- Running the *INSTALL-CA [GROUP]* command from the SCEP menu.
- Using the **auto-install** feature (this is explained in greater detail in group configuration options).

The install process must have the domain name configured in the group SCEP menu through the *DOMAIN-NAME* command or in the IP menu through the *DNS-DOMAIN-NAME* command.

Enroll

You need to install the user certificate after installing the CA certificate. The device will need an RSA key before it can contact the SCEP server to request a certificate. This key should have been generated earlier but will be generated automatically (as explained in the commands section) if it doesn't already exist. Basically, the device encapsulates a CSR request and sends it to the server which validates the request, generates the corresponding certificate and returns it, also encapsulated.

You can have someone manually installing the user certificate through the *enroll* process, or you can configure automatic installation. With automatic installation, the device attempts to obtain a certificate when it doesn't have one or when one has expired.

View status

The certificates that are obtained are automatically installed on the device. You can view them from the CERT menu. Certificates have the name assigned in the SCEP configuration and are associated with a template, as discussed in the section on *Obtaining certificates through CSR* on page 132. To check the general status of the configuration, use the *LIST TEMPLATE ALL* command from the IPsec menu. To list the status of configured groups, use the *LIST* command from the SCEP monitoring menu.

CONFIGURATION

The device allows you to configure multiple SCEP servers with corresponding parameters within groups that can be added to the configuration. To enter the SCEP menu, run the *SCEP* command from the IPsec CERT menu.

Command	Function
CA-CHAIN-INSTALL	Installs the certificate chain up to the root CA.
CANCEL-PENDING	Cancels pending operations.
CAPABILITIES	Displays commands supported by the server.
ENROLL	Runs the enroll protocol for a SCEP group.
GROUP	Creates or enters a SCEP group configuration.
INSTALL-CA	Runs the install protocol for a SCEP group.
NEXT-CA-INSTALL	Installs the renewed CA certificates.
NO	Deletes an option or restores it to its default value.
EXIT	Exits the SCEP setup menu.

“CA-CHAIN-INSTALL [group]”

Runs a SCEP group *GetCACertChain* request.

This command should be used when the *INSTALL* command fails to install the certificate chain up to the root CA. With some older servers, the *INSTALL* command will only give the root CA certificate and not the whole chain. We don't recommend using this command unless this happens.

Example:

```
hub1 SCEP config$ca-chain-install 1
Installing CA certificate...
Opening...
172.24.75.193...
Sending Query. Waiting Answer...
Certificate name: BINTEC.CER
Fingerprint: 79 34 C8 FA 4C 21 AD 82 45 5F 4C 51 C5 D7 9A 64
Do you accept the certificate received(Yes/No)? y
Saving CA certificate bintec
Version                   : V3
Serial Number             : 009C 78A9 B776 C6E8 9E46 CF33 8889 E779
Algorithm Identifier      : SHA1 With RSA
Issuer:
  CN (Common Name        ):
**************************************
Do not forget to save the configuration!
**************************************
```

“CANCEL-PENDING [group]”

This command should be used to cancel pending operations in a group that has already been configured.

Example:

```
hub1 SCEP config$cancel-pending ?
   <0..4294967295>    group number
hub1 SCEP config$cancel-pending 1
```

Command history:

Release	Modification
11.01.03	This command was introduced as of version 11.01.03.

“CAPABILITIES [group]”

Runs a SCEP group *GetCACaps* request, which displays the features supported by the server. This command was introduced in 2008. Servers with versions prior to this date will give an incorrect response and you will get a parsing/ command error (or similar) in the console.

Example:

```
hub1 SCEP config$capabilities 1
Opening...
172.24.75.193...
Sending Query. Waiting Answer...
"GetNextCACert"
"POSTPKIOperation"
```

“ENROLL [group]”

Runs a SCEP group *enroll* process. This forces the process to run independently of the AUTOENROLLMENT parameter configured in the group.

Upon receiving the certificate, the certificate *footprint* is displayed for user approval. The footprint is the certificate's MD5 hash.

Remember to save the configuration to preserve the certificate and generated keys the next time the device is rebooted.

Example:

```
SCEP config$enroll 1
Building CSR...
Ciphering enveloped data...
Building signature...
Opening...
172.24.75.193...
Sending Query. Waiting Answer...
Certificate name: ROUTER.CER
Fingerprint: 42 A4 8F 61 E4 1D 39 91 7B 34 0B EA C6 09 B3 47
Do you accept the certificate received(Yes/No)? y
Saving CA certificate router
Version                   : V3
Serial Number             : 1C
Algorithm Identifier      : MD5 With RSA
Issuer:
  CN (Common Name        ):
**************************************
Do not forget to save the configuration!
**************************************
```

“GROUP [group]”

Creates or enters a SCEP group configuration.

Example:

```
SCEP config$group 1
-- Scep group user configuration --
SCEP group 1 config$
```

“INSTALL-CA [group]”

Runs a SCEP group *install* process.

Upon receiving the certificate, the certificate *footprint* is displayed for user approval. The footprint is the certificate's MD5 hash.

Remember to save the configuration to preserve the certificate and generated keys the next time the device is rebooted.

Example:

```
hub1 SCEP config$install-ca 1
Installing CA certificate...
Opening...
172.24.75.193...
Sending Query. Waiting Answer...
Certificate name: BINTEC.CER
Fingerprint: 79 34 C8 FA 4C 21 AD 82 45 5F 4C 51 C5 D7 9A 64
Do you accept the certificate received(Yes/No)? y
Saving CA certificate bintec
Version                   : V3
Serial Number             : 009C 78A9 B776 C6E8 9E46 CF33 8889 E779
Algorithm Identifier      : SHA1 With RSA
Issuer:
  CN (Common Name        ):
**************************************
Do not forget to save the configuration!
**************************************
```

“NEXT-CA-INSTALL [group]”

Runs a SCEP group *GetNextCACert* request allowing you to obtain new CA certificates before the old ones expire. This command was introduced in 2008. Servers with versions prior to this date will give an incorrect response and you will get a parsing/command error (or similar) in the console.

Example:

```
hub1 SCEP config$next-ca-install 1
Installing CA certificate...
Opening...
172.24.75.193...
Sending Query. Waiting Answer...
Certificate name: BINTEC.CER
Fingerprint: 79 34 C8 FA 4C 21 AD 82 45 5F 4C 51 C5 D7 9A 64
Do you accept the certificate received(Yes/No)? y
Saving CA certificate bintec
Version                   : V3
Serial Number             : 009C 78A9 B776 C6E8 9E46 CF33 8889 E779
Algorithm Identifier      : SHA1 With RSA
Issuer:
  CN (Common Name        ):
**************************************
Do not forget to save the configuration!
**************************************
```

Command history:

Release	Modification
11.01.03	The "*next-ca-install*" command is obsolete as of version 11.01.03.

SCEP GROUP CONFIGURATION

Command	Function
ALTERNATIVE-NAME	Alternative name for a CSR request.
AUTOENROLLMENT	Configures auto enrollment.
CA-CERT-NAME	Name assigned to the associated CA certificate.
CA-FINGERPRINT	CA certificate fingerprint.
CA-ISSUER-AS-RECIPIENT	Uses CA issuer name as recipient info.
CGI-PATH	URL path.
CHALLENGE-PASSWORD	CSR password.
DEBUG	Debugging mode.
DOMAIN-NAME	CA domain name.
GENERATEKEY	Generates an RSA key if the configured key can’t be found.
IP-ADDRESS	Adds the configured IP to the CSR *subject*.
PORT	Server listening port.
REGENERATEKEY	Regenerates the RSA key for each enrollment.
ROLLOVER	Enables rollover.
RSA-KEY-ID	RSA key ID.
RSA-KEY-LENGTH	RSA key length.
SERIAL-NUMBER	Adds the device's serial number to the CSR *subject*.
SIGNER-NAME	Message signer name.
SOURCE-ADDRESS	Source IP address.
SUBJECT-NAME	CSR subject-name.
URL	Server URL.
USER-CERT-NAME	Name assigned to the associated user certificate.
VRF	VRF where SCEP operates.
NO	Deletes an option or restores it to its default value.
EXIT	Exits the SCEP setup menu.

“ALTERNATIVE-NAME [Alternative-name]”

Configures the *alternative-name* attribute that forms part of the CSR generated in the *enroll* process.

Use the *NO ALTERNATIVE-NAME* command to delete the *alternative-name*.

Example:

```
SCEP group 1 config$alternative-name ?
  <word>    Text
SCEP group 1 config$alternative-name bintec.randd.de
```

“AUTOENROLLMENT [time before expiration] [RETRY-PERIOD <value>]”

Configures the automatic mode. When the device is in automatic mode, it runs the *enroll* process when the following occurs:

- When the certificate identified by name using the *USER-CERT-NAME* command does not exist.
- When the certificate identified using the *USER-CERT-NAME* command has less than double the configured expiration time left.

The device periodically checks whether the above circumstances have occurred. Checks are performed every minute (by default) but you can configure the *RETRY-PERIOD* to change their frequency.

Before you can run the *enroll* process, you must perform the installation. Please remember that the installation is not an automatic process.

Run *NO AUTOENROLLMENT* to disable automatic enrollment.

This procedure is disabled by default.

Example:

```
SCEP group 1 config$autoenrollment ?
  <1m..52w>    Time value
SCEP group 1 config$autoenrollment 30m ?
  retry-period    Wait period between certificate request retries
  <cr>
SCEP group 1 config$autoenrollment 30m retry-period ?
  <1m..52w>    Retry time in minutes
SCEP group 1 config$autoenrollment 30m retry-period 4m
```

Command history:

Release	Modification
10.08.43	As of version 10.08.43, the value of this command is the amount of time before expiration rather than a period of time.
10.09.25	As of version 10.09.25, the value of this command is the amount of time before expiration rather than a period of time.
10.09.24.20.07	The *RETRY-PERIOD* option was introduced as of version 10.09.24.20.07.
11.00.00.02.08	As of version 11.00.00.02.08, the value of this command is the amount of time before expiration rather than a period of time.
11.00.04	As of version 11.00.04, the value of this command is the amount of time before expiration rather than a period of time. The *RETRY-PERIOD* option was also introduced.
11.01.00	As of version 11.01.00, the value of this command is the amount of time before expiration rather than a period of time.The *RETRY-PERIOD* option was also introduced.

“CA-CERT-NAME [name]”

Identifies the name that the CA certificate has once the certificate has been obtained, or the name it already has if it has already been loaded.

The default value for this command is “bintec”. Run the *NO CA-CERT-NAME* command to restore the default value.

"CA-FINGERPRINT [type] [fingerprint]"

Configures a fingerprint in MD5 or SHA1 format for the CA certificate. Both types can be used simultaneously. For an install operation to be successful, the configured fingerprint must match the received fingerprint. If said print doesn't match, the install operation stops and the certificate won't be saved.

If both types of fingerprint are configured, the two must match.

```
SCEP group 1 config$ca-fingerprint ?
```

```
  md5     Fingerprint in md5 format
  sha1    Fingerprint in sha1 format
SCEP group 1 config$ca-fingerprint
```

```
SCEP group 1 config$ca-fingerprint md5 ?
  <47 chars>    MD5 fingerprint, 16 bytes in hex format with ':' separator
SCEP group 1 config$ca-fingerprint md5
```

```
SCEP group 1 config$ca-fingerprint sha1 ?
  <59 chars>    SHA1 fingerprint, 20 bytes in hex format with ':' separator
SCEP group 1 config$ca-fingerprint sha1
```

AUTO-INSTALL FEATURE:

When a ca-fingerprint has been configured, the router tries to automatically install a CA certificate if the following conditions are met:

- Autoenroll is enabled (configured with any autoenroll period).
- The currently saved CA certificate does not exist or has expired.

In this case, the auto-install process is initiated and a CA certificate retrieved. This CA certificate is only saved if the configured fingerprints match the configured CA certificate fingerprints.

Once the CA installation has completed successfully, an auto-enroll operation executes. The CA and router certificates are then automatically installed.

Example 1:

```
SCEP group 1 config$ca-fingerprint md5 12:34:56:78:90:AB:CD:EF:12:34:56:78:90:AB:CD:EF
```

Example 1 shows the MD5 fingerprint format: 16 bytes represented in hexadecimal format using the colon (:) symbol as separator.

Example 2:

```
SCEP group 1 config$ca-fingerprint sha1 12:34:56:78:90:AB:CD:EF:12:34:56:78:90:AB:CD:EF:12:34:56:78
```

Example 2 shows the SHA1 fingerprint: 20 bytes represented in hexadecimal format using the colon symbol (:) as separator.

Run *NO CA-FINGERPRINT MD5* or *NO CA-FINGERPRINT SHA1* to remove the preset fingerprints. Note that the auto-install feature can't execute if the *NO CA-FINGERPRINT* command is configured.

Command history:

Release	Modification
11.01.02	This command was introduced as of version 11.01.02.

“CA-ISSUER-AS-RECIPIENT”

Uses the issuer name of the CA for the recipient information in the *enroll* process.

By default, the CA subject name is used for the recipient information. Run the *NO CA-ISSUER-AS-RECIPIENT* command to restore default behavior.

Command history:

Release	Modification
10.08.43	This command was introduced as of version 10.08.43.
10.09.25	This command was introduced as of version 10.09.25.
11.00.00.02.08	This command was introduced as of version 11.00.00.02.08.
11.00.04	This command was introduced as of version 11.00.04.
11.01.00	This command was introduced as of version 11.01.00.

“CGI-PATH [url-path]”

Specifies the server URL path (usually a CGI script). Default is “/cgi-bin/pkiclient.exe”.

```
SCEP group 1 config$cgi-path ?
  <word>    Text
```

```
SCEP group 1 config$cgi-path /~pkii/pkiclient.php
```

Run the *NO CGI-PATH* command to restore the default value.

“CHALLENGE-PASSWORD [password] | [CIPHERED <password>]”

Configures the password attribute that forms part of the CSR generated in the *enroll* process. You can configure a password in ciphered format using the *CIPHERED* option.

Example 1:

```
SCEP group 1 config$challenge-password ?
  <1..256 chars>    Key value
  ciphered          Ciphered password
SCEP group 1 config$challenge-password hello
```

Example 2:

```
SCEP group 1 config$challenge-password ciphered ?
  <1..531 chars>    Key value
SCEP group 1 config$challenge-password ciphered 0x794B61FA286B1222
```

Run the *NO CHALLENGE-PASSWORD* command to delete the password attribute.

Command history:

Release	Modification
10.09.24.20.07	The *CIPHERED* option was introduced as of version 10.09.24.20.07.
11.00.04	The *CIPHERED* option was introduced as of version 11.00.04.
11.01.00	The *CIPHERED* option was introduced as of version 11.01.00.

“DEBUG”

This command is only effective for operations forced from the console. With this enabled, any requests sent to the server are printed for further analysis. Use the *NO DEBUG* command to disable this function.

“DOMAIN-NAME”

Sets the CA domain name. Where this parameter is not configured, the CA domain name will be the name configured in the IP with the *DNS-DOMAIN-NAME* command.

If this command is not set and there is no DNS-DOMAIN-NAME, the *install* process will end with an error.

Run the *NO DOMAIN-NAME* command to delete the CA domain name.

“GENERATEKEY”

Makes the *enroll* process generate an RSA key when the one configured using the *RSA-KEY-ID* command cannot be found.

If a key configured with the *RSA-KEY-ID* command exists, then this command has no effect.

If there is no key configured with the *RSA-KEY-ID* command and this command is not enabled, the *enroll* process ends with an error.

Please see the *RSA-KEY-ID* and *REGENERATEKEY* commands.

This command is disabled by default. Run the *NO GENERATEKEY* command to restore the default behavior.

“IP-ADDRESS [IP ADDR/INTERFACE]”

If this command is configured, the IP is included in the CSR *subject* generated in the *enrollment* process. You can configure an IP address or an interface. If you configure an interface, the interface's primary IP is used. The IP is encoded in the *unstructuredAddress* attribute. By default, the IP is not included in the CSR.

Example:

```
SCEP group 1 config$ip-address ?
  <a.b.c.d>      Ipv4 format
  <interface>    Interface name
SCEP group 1 config$ip-address ethernet0/0
```

Run the *NO IP-ADDRESS* command to delete the IP address.

“PORT”

Specifies the server listening port. Default is 80 (HTTP). Run the *NO PORT* command to restore the default value.

“REGENERATEKEY”

Generates an RSA key every time the *enroll* process is run, provided that the key configured with the *RSA-KEY-ID* command exists. It is important to note that if the RSA key specified using the *RSA-KEY-ID* command does not exist and the *GENERATEKEY* command is not enabled, the *enroll* process ends with an error regardless of this command's value.

Please see the *RSA-KEY-ID* and *GENERATEKEY* commands.

This command is disabled by default. Run the *NO REGENERATEKEY* command to restore the default behavior.

“ROLLOVER [time before expiration] [RETRY-PERIOD <value>]”

The device periodically checks whether the above circumstances have occurred. By default, checks are performed every minute but you can configure the *RETRY-PERIOD* to change their frequency.

Run *NO ROLLOVER* to disable.

This procedure is disabled by default.

Example:

```
SCEP group 1 config$rollover ?
  <1m..52w>   CEime value
SCEP group 1 config$rollover 30m ?
  retry-period    Wait period between rollover retries
  <cr>
SCEP group 1 config$rollover 30m retry-period ?
  <1m..52w>    Retry time in minutes
SCEP group 1 config$rollover 30m retry-period 4m
```

Command history:

Release	Modification
11.01.03	This command was introduced as of version 11.01.03.

“RSA-KEY-ID [RSA key id]”

Specifies the RSA key used in this SCEP group. RSA keys are generated in the IPSEC menu using the *KEY RSA GENERATE* command. The device will use the *enroll* process to generate an RSA key if the configured key cannot be found or if this command is disabled, but it will only do so if the *GENERATEKEY* command is enabled.

Please see the *GENERATEKEY* command and the *REGENERATEKEY* commands.

This command is disabled by default. Run the *NO RSA-KEY-ID* command to restore the default behavior.

As an example, the *enroll* process has the following effects in these configurations.

- (no rsa-key-id) && (generate):

  Generates a key in the first *enroll* process that is retained in successive *enrollments*.
- (no rsa-key-id) && (generate) && (regenerate):

  Generates a key in the first *enroll* process that is renewed in successive *enrollments*.
- (rsa-key-id 3):

  If key id 3 exists, it is used in the first and successive *enroll* processes.

  If it does not exist, then the *enroll* process ends with an error.
- (rsa-key-id 3) && (regenerate):

  If key id 3 exists, it is regenerated in successive *enroll* processes.

  If it does not exist, then the *enroll* process ends with an error.

“RSA-KEY-LENGTH [length in bits]”

Specifies the length (in bits) of the RSA key automatically generated. This command has no effect if the key specified through the *RSA-KEY-ID* command already exists.

If the key specified through the *RSA-KEY-ID* command does not exist, the keys that are automatically generated in *enroll* process will have the length that is specified here.

If the key specified through the *RSA-KEY-ID* command does not exist and this command is not configured, the keys that are automatically generated in the *enroll* process will have the CA certificate module length.

Run the *NO RSA-KEY-LENGTH* command to delete the configured length.

“SERIAL-NUMBER”

If this command is configured, the device's serial number is included in the CSR *subject* generated in the *enroll* process. The serial number is encoded in the *serialNumber* attribute. By default, the serial number is not included in the CSR. Run the *NO SERIAL-NUMBER* command to restore the default behavior.

“SIGNER-NAME [name]”

Specifies the *signer-name* identifier attached to the message sent in the *enroll* process.

Example:

```
SCEP group 1 config$signer-name examplesignername
SCEP group 1 config$
```

Run **NO SIGNER-NAME** to delete the configured value.

When this command isn't configured, the *signer-name* is obtained from the domain name, configured through the *DNS-DOMAIN-NAME* command from the IP protocol menu, preceded by the device name, configured through the *SET-HOSTNAME* command from the configuration root menu, and a period.

Example:

DNS-DOMAIN-NAME = bintec.de

HOSTNAME = hostname

The signer-name in this case is: hostname.bintec.de

When this command is not configured and there is no HOSTNAME or DNS-DOMAIN-NAME, the *enroll* process ends with an error.

Run **NO SIGNER-NAME** to restore the default value.

“SOURCE-ADDRESS [IP ADDR/INTERFACE]”

The *SOURCE-ADDRESS* command specifies the interface address that will be used as source address for all outgoing TCP connections. The source can be an IP address or an interface. If you configure an interface, the interface's primary IP is used. If you use the *VRF* command to configure a VRF and SCEP is being used, the interface's VRF must be the same as the VRF you set when configuring the source address as an interface. Otherwise, you'll get an error message.

Example:

```
SCEP group 1 config$source-address ?
  <a.b.c.d>      Interface IP address
  <interface>    Interface name
SCEP group 1 config$source-address ethernet0/0
```

Use the *NO SOURCE-ADDRESS* command to delete the source IP address.

Command history:

Release	Modification
10.09.20	This command was introduced as of version 10.09.20.
11.00.02	This command was introduced as of version 11.00.02.

“SUBJECT-NAME [name]”

Configures the CSR *subject-name* attribute generated in the *enroll* process.

To configure said *subject-name*, enter the different fields as a character string separated by commas and in X500 format, as shown below:

X500 format: " **C** = Country**, L** = Locality**, ST** = State or Province **, O** = Organization,

OU = Organization Unit**, CN** = Common Name**, E** = Email ... "

Example:

```
SCEP group 1 config>subject-name ?
  <1..250 chars>    Subject-name: " C=Country, L=Locality, ST=State/Province,
                    O=Organization, OU=OrgUnit, CN=CommonName, E=Email,
                    DC=DomainComponent... "
SCEP group 1 config>subject-name "c = es, l = madrid, o = bintec, ou = imasd, ou = planta1"
SCEP group config>subject-name "cn = bintec.randd.de, e = prueba@bintec.de"
SCEP group config>
```

Note

1. Make sure you enter the backslash character (\) before entering an equals sign (=), backslash (\) or comma (,) in any of the fields.

2. Inverted commas (“) are not permitted in any of the fields.

When this command is not configured, the CSR *subject-name* is obtained from the domain name (configured through the *DNS-DOMAIN-NAME* command in the IP protocol menu) preceded by the device name (configured through the *SET HOSTNAME* command from the configuration root menu) and a period.

Example:

DNS-DOMAIN-NAME = bintec.de

HOSTNAME = hostname

The subject-name in this case is: hostname.bintec.de

When this command is not configured and there is no *HOSTNAME* or *DNS-DOMAIN-NAME*, the *enroll* process will end with an error.

Use *NO SUBJECT-NAME* to delete the configured value.

“URL [url]”

Specifies the URL that the server listens on.

Examples:

```
SCEP group 1 config>url ca.bintec.de
SCEP group 1 config>url 172.24.78.78
```

If the specified URL is not an IP address, the device will require a DNS server capable of resolving domain names.

Use the *NO URL* to delete the configured URL.

“USER-CERT-NAME [name]”

Identifies the name of the user certificate once it has been obtained, or the name of the CA certificate if it has already been loaded.

The default value for this command is "router". Use *NO USER-CERT-NAME* to restore the default value.

“VRF [vrf_tag]”

Specifies the VRF where SCEP is working. If the *SOURCE-ADDRESS* command is configured as an interface, the interface's VRF must be the same or the device will return an error.

By default, no VRF is configured and VRF_MAIN is used. Use *NO VRF* to restore the default value.

Command history:

Release	Modification
11.00.04.20.04	This option was introduced as of version 11.00.04.20.04.
11.00.05	This option was introduced as of version 11.00.05.
11.01.01	This option was introduced as of version 11.01.01.

2.5.6 Certificate revocation list (CRL)

In some situations, such as when there is a security risk, change of name or device, you might need to cancel or invalidate a certificate. You can check whether a certificate is still valid by consulting the list of invalidated certificates called the *Certificate Revocation List* (CRL).

A router can obtain a CRL from a lightweight directory access protocol (LDAP) server, usually located in the CA itself. The IPsec LDAP menu allows you to configure four servers from which to download the CRL. Afterwards, the server is assigned a *template*. Once the device has obtained the CRL, it stores it in the non-volatile memory to prevent it from being deleted the next time the device reboots.

The device supports delta CRLs. These are CRLs that only provide information about certificates whose statuses have changed since the issuance of a previous CRL.

You can also download the CRL and install it in the device's non-volatile memory (via FTP, for example).

In order to configure the CRL, you must define the search parameters to access the server and enable the CRL in the template. To configure the search parameters, use the IPsec *LDAP* command described in the section on *IPsec LDAP command* on page 147. To enable the CRL use the *CRL* command from the *TEMPLATE* menu of IPsec as explained in the section on *Template CRL command* on page 148.

2.5.6.1 IPsec LDAP command

The *LDAP* command from the IPsec menu has the following options:

Command	Function
SERVER	Configures the LDAP server parameters.
TIMER	CRL query period in the LDAP server.

“LDAP SERVER [ID] DEFAULT”

Configures the LDAP server listening port. The default is port 389.

Example:

```
IPSec config>ldap server 1 default
```

Command history:

Release	Modification
11.00.05	The *LDAP server default* command is obsolete as of version 11.00.05.
11.01.00	The *LDAP server default* command is obsolete as of version 11.01.00.

“LDAP SERVER [ID] DESTINATION ADDRESS [IP ADDR]”

Configures the LDAP server IP address or the domain name.

You can choose to let the device get the server address by searching the CA certificate *CRL Distribution Points* extension. To achieve this behavior, you need to configure the *USE-CA-SUBJ-AS-DN* option in the *TEMPLATE*, as indicated in the section on *Template CRL command* on page 148.

Example:

```
IPSec config>ldap server 1 destination address ldap.bintec.de
```

“LDAP SERVER [ID] DESTINATION PORT [PORT]”

Configures the LDAP server listening port. The default is port 389.

Example:

```
IPSec config>ldap server 1 destination port 370
```

“LDAP SERVER [ID] SOURCE-ADDRESS [IP ADDR]”

Configures the source IP address used in LDAP server queries.

Example:

```
IPSec config>ldap server 1 source-address 2.2.2.2
```

“LDAP SERVER [ID] DN [Distinguished Name]”

Configures the *distinguished name* (DN) attribute used in LDAP server queries.

You can choose to let the device obtain the DN by searching the CA certificate *CRL Distribution Points* extension. To achieve this behavior, you need to configure the *USE-CA-SUBJ-AS-DN* command in the *TEMPLATE*, as indicated in *Template CRL command* on page 148.

Example:

```
IPSec config>ldap server 1 dn "ou=For Test Purposes Only,o=Bintec"
```

“LDAP SERVER [ID] AUTHENTICATION [String]”

Configures the simple authentication string used in LDAP server queries.

Example:

```
IPSec config>ldap server 1 authentication "bintec"
```

“LDAP SERVER [ID] NAME-AUTH [String]”

Configures the simple authentication name used in LDAP server queries.

Example:

```
IPSec config>ldap server 1 name-auth "user@bintec.de"
```

“LIST LDAP SERVER”

Displays a list of configured servers. We would obtain the following if we use what we configured in the previous examples:

Example:

```
IPSec config>list ldap server
LDAP Server 1
    destination: ldap.bintec.de
    destination port: 370
    source address: 2.2.2.2
    dn: ou=For Test Purposes Only,o=Bintec
    name used for authentication: user@bintec.de
    authentication: bintec
```

“LDAP TIMER [SECONDS]”

Sets the period of time between CRL lookups. The default value is one day.

The CRL has several optional fields: *next update* indicates the date of the next CRL update, and *next publish* indicates the date the next CRL/delta CRL will be issued. The update is done on the date indicated in the *next update* or *next publish* fields, or upon expiry of the period scheduled between lookups, whichever occurs first.

Example:

```
IPSec config>ldap timer 2d
IPSec config>list ldap timer

Period to check LDAP servers: 48h0m0s
```

2.5.6.1.1 Attributes

The default search attribute used in LDAP to obtain the CRL in IPsec is as follows:

certificaterevocationlist;binary

You can select another attribute by changing the LDAP global configuration, as explained in the *bintec Dm790-I LDAP Protocol* manual. When you configure an attribute in the LDAP global configuration, you can use it in the CRL search rather than in the default.

The other search parameters used are obtained on the basis of those configured in the LDAP global configuration.

2.5.6.2 Template CRL command

As explained in the section on *ISAKMP template parameters* on page 25, setting the *TEMPLATE [ID] IKE METHOD* to *RSA* when configuring ISAKMP templates, gives you the *TEMPLATE [ID] IKE CRL* command with the following options:

Command	Function
OPTIONAL	Continuous even if the CRL is not available.
ALWAYS	CRL must always be available.
LDAP-SERVER	Assigns an IPsec LDAP server.
USE-CA-SUBJ-AS-DN	Uses the CA subject as the CRL DN.

“TEMPLATE [ID] IKE CRL OPTIONAL”

If a certificate cannot be validated using a CRL lookup because the CRL is not available, the device will assume that the certificate has not been revoked and will use it. The *OPTIONAL* or *ALWAYS* command must be enabled for the device to use the CRL.

Example:

```
IPSec config>template 22 ike crl optional
```

Use *TEMPLATE [ID] IKE NO CRL* to disable this option.

“TEMPLATE [ID] IKE CRL ALWAYS”

If a certificate cannot be validated using a CRL lookup because the CRL is not available, the device will assume that the certificate may have been revoked and will not use it. The *OPTIONAL* or *ALWAYS* command must be enabled for the device to use the CRL.

Example:

```
IPSec config>template 22 ike crl always
```

Use *TEMPLATE [ID] IKE NO CRL* to disable this option.

“TEMPLATE [ID] IKE CRL LDAP-SERVER [ID]”

This command is used to assign an LDAP server configured in IPsec to the template. When you need to search for a CRL using LDAP, the connection parameters will be taken from the server assigned to this command.

Example:

```
IPSec config>template 22 ike crl ldap-server 2
```

Use *TEMPLATE [ID] IKE NO CRL* to dissociate the LDAP server.

“TEMPLATE [ID] IKE CRL USE-CA-SUBJ-AS-DN”

When the device starts searching for the CRL using LDAP, it tries to get the server address and DN from the CA certificate *CRL Distribution Points* extension.

- If the DN is not found, it is taken from the CA certificate *subject* field.
- If the server address is not found, the one configured in the associated LDAP server is used.

When you want to use a different DN (i.e., not the CA *subject*) or you don't want to use the address of the server located between the CA certificate extensions, you must set the *NO USE-CA-SUBJ-AS-DN* command and configure the DA and IP address in the associated LDAP server (as indicated in the section describing the IPsec "*LDAP*" command).

Example:

The DN is the same as the CA subject.

```
IPSec config>template 22 ike crl use-ca-subj-as-dn
```

Example:

The DN is different from the CA subject.

```
IPSec config>template 22 ike crl ldap-server 2
IPSec config>template 22 ike crl no use-ca-subj-as-dn
IPSec config>ldap server 2 destination address 81.11.11.121
IPSec config>ldap server 2 dn "CN=Test,C=ES"
```

Use the *LIST TEMPLATE ALL* command to display the configuration status:

```
IPSec config>list template all
TEMPLATES
22 isakmp 3DES MD5  DES=0.0.0.0
```

```
   LifeTime:1h0m0s
   IKE MAIN
   RSA SIGNATURE
      CA      :VRSGNCA.CER
               -ou=For Test Purposes Only,o=Bintec              -Without Private Key
               -   Signature ok.
      CRL     : VRSGNCA.CRL
               -Search of CRL by Subject of CA failed, VRSGNCA.CER
               -ou=For Test Purposes Only,o=Bintec              -Last update 0h12m31s
               -Next update in 0h0m40s
               -Number of items 5
               -ALWAYS enabled => CA must always be available
               -LDAP server number 3
      USER    :
   fqdn ID TYPE
   OAKLEY GROUP 1
```

There are a number of operations that you can perform with the CRL lists, such as *LIST*, *PRINT*, *DELETE*, *LOAD*, and *UNLOAD*. Running commands in this menu will not change the configuration.

The *TIME-TO-EXPIRE* command allows you to temporarily bring forward a CRL search to take place at a time other than the scheduled time.

```
IPSec config>cert
-- Cert user configuration --
CERTIFICATES config>crl
-- CRL user configuration --
CRL config>?
  delete            Delete a CRL
  list              List CRLs
  load              Load a CRL
  print             Print a CRL
  time-to-expire    Configure the time to expire of a CRL
  unload            Unload a CRL
  exit
CRL config>print all

Name: VRSGNCA.CRLVersion                  : V2
Algorithm Identifier      : SHA1 With RSA
DN                        : ou=For Test Purposes Only,o=Bintec
This Update               : Wed Oct 29 08:00:07 2008

Next Update               : Thu Oct 30 08:00:07 2008
Last update               : 0h18m4s ago
Next update in            : 0h0m45s
Ldap status               : LDAPSTATUS_NOSUCHOBJECT
Number of items           : 5

Signature                 :
  Signature Algorithm     : SHA1 With RSA
  Signature Data Info     : 2048 Bits.
  Signature Data          :
    5C33 20CC FACA BD65 76AB 3FDB 4786 E620 FC0E 0DA4 B934 E745 2ACC 2453 9177
    9DCB 07D1 1FEB C6A1 0812 896E 6042 1DC2 A94D 85B6 60FA 4656 C07D 22BD 5C37
    6907 4765 E6F7 1CCC 6F44 B651 68F3 AC39 6886 9A79 13FC FAD4 8F79 04BF 69CA
    F68F 08BA A85D AB22 5BD2 1BA3 8B96 8961 380D 2B0C E157 274B 72C1 FB92 71D6
    0E3D 757A FAB5 A83B E903 9B13 A225 0183 2381 629E 6DE1 C099 2841 AC9E 6915
    6A05 25B1 0133 804D 07D1 A8D1 E94E 74C5 6745 0D65 0DA4 2776 215A 84B7 E0F8
    2350 EBD1 00DE 1B62 DCE7 0288 8A55 9199 03AB DFBF 185F 40F0 D12D 9C15 A6C2
    8CF1 9BD9 4329 81E0 3664 26A5 C831 F13F 9107 929A D2C1 8ECA 6F11 6D5F 4D4A
    15DB FBED 3C1A 7509 6464 DC78 C203 3935
```

2.5.7 PKCS #12. Personal Information Exchange

PKCS #12 is a standard that uses an established syntax to provide a safe means of exchanging personal identity information. Some of the data that can be exchanged include RSA passwords and certificates.

The standard is basically implemented in the device with two command groups:

- Commands for **importing** certificates and RSA passwords.
- Commands for **exporting** certificates and RSA passwords.

As explained in the section on *CERT menu* on page 128, the CERT menu is located in the IPsec menu. The *CERTIFICATE* command within the CERT menu has the following options relating to the PKCS12 standard:

Command	Function
PKCS12 IMPORT	Uploads a P12 file from a configuration disk.
BASE64	Uploads a certificate and password in base64 format from the console.
PKCS12 EXPORT [CHAIN]	Exports a certificate and the associated RSA password.

“CERTIFICATE [CertFile] PKCS12 IMPORT”

Allows you to load and process a PKCS12 file with a “P12” extension from the device’s static memory (disk).

The P12 file is protected at all times by a password, which is requested by the device when the command is entered.

Once the command has executed successfully, the certificates contained in the file and an RSA password are uploaded to the device's configuration. At the same time, the file is deleted from the device’s static memory (disk) for security reasons.

```
router1 IPSec config>list key rsa
0 rsakey entries
Id.           Date.         Len         Certificate Authority (CA)           Cert sn.
router1 IPSec config>cert

-- Cert user configuration --
router1 CERTIFICATES config>list config-certificates
No Certificates configured
router1 CERTIFICATES config>
router1 CERTIFICATES config>list disk-certificates
 A:                 FULLROUT.P12       3532   04/11/13   12:11   Flash
 A:                   DEBIAN.P12       2445   04/21/13   12:56   Flash
router1 CERTIFICATES config>certificate DEBIAN pkcs12 import
Enter password:************
router1 CERTIFICATES config>list config-certificates

                 DEBIAN.CER

router 1 CERTIFICATES config>exit
router1 IPSec config>list key rsa
1 rsakey entries
Id.           Date.         Len         Certificate Authority (CA)           Cert sn.

1   04/13/13  09:36:08   2048                 00E6 815F 5421 A22E B3
```

“CERTIFICATE [CertFile] BASE64”

Allows you to enter a PKCS #12 file in base64 format. As with the previous configuration, the certificates and RSA key are saved to the configuration once the command has executed successfully.

```
router1 CERTIFICATES config>certificate debian64 base64

Introduce the Certificate (Base 64 format)
Enter <cr> to escape
MIIJiQIBAzCCCU8GCSqGSIb3DQEHAaCCCUAEggk8MIIJODCCA+8GCSqGSIb3DQEH
BqCCA+AwggPcAgEAMIID1QYJKoZIhvcNAQcBMBwGCiqGSIb3DQEMAQYwDgQIU4qt
(..)
fxazH9E5mqZdVuuhDs8DB2D7FPgBJJzgVnrpBrqPZFM3murz6T5y1kPGwMkQCsRj
NVUXWrhc0RJ8bANxTZu1iR1QbOFKs4fEeM9zguob3thDrPze8lIksi2LQ2vslyWY
MtVfmq6Rt6aR+IdXWlDTkPggjVdm4d+1lUbETMpq8TRbMICni8SWmlvuENS0WAQ+
G4aIwqQxxPjX2ayci994Oljx5Nittc7QC49/ucSPhHJkcC3tKI+UYLJltsaJK0j5
(..)
/iuOlI7dsHUk0g5psrM9xXNY1JgUULw3qK1zPfjgDbMItWhXybtmgddqSGNQNO5B
Yyufl9cCOxWcKx6tC8NeMrGh8OadD1962d7D3qVVgCD9gJWNcP9WUfM+mNdsPuHe
Ru3fj6b7jVv9rFpgS3tuugDWb7diyo82NB8pauC9nBC4aXp9z3RT054A0KRYjCGp
(..)
khwUMSUwIwYJKoZIhvcNAQkVMRYEFDPB8xY8OkNISoRe+e2h4SLinl/iMDEwITAJ
BgUrDgMCGgUABBSLhDDe+e4RyBzIaTp/YVVFoe/h+QQIrUvhs4YmwJoCAggA
Enter password:************
```

“CERTIFICATE [CertFile] PKCS12 EXPORT [CHAIN]”

Allows the device to export a certificate and its associated RSA key. The certificate and key are exported in base64 format. This appears on screen when the command is run.

The device requests a key to protect the content.

The device tries to export the entire user certificate chain when the CHAIN option is present.

```
router1 CERTIFICATES config>certificate debian pkcs12 export
Enter password:************
Confirm password:************
MIIHSAIBAzCCBwQGCSqGSIb3DQEHAaCCBvUEggbxMIIG7TCCAtYGCSqGSIb3DQEH
AaCCAscEggLDMIICvzCCArsGCyqGSIb3DQEMCgECoIICZjCCAmIwHAYKKoZIhvcN
AQwBAzAOBAh0IcKZinXr2AICB9AEggJA0HWp52Cwx6hWni8xOfQPobG9PjY8eQKF
(..)
nV/uAj8z2gH4TsKxjKFyoJj9rtGu+XskXcPrkLqbAPXfArZW4cpBTG7j3P8c9Lv1
syv3IKMx+b8B/lafyYEVXNOPucQ6Jnt2X8o1p+iTXkULTxk+UWCYpnhPwHOPzRTX
ogy4pBwJWXQxRxdf1+L7InxTEAoeIjLXrJ3jX48YgNgMgp0gPWw4txfS90HAYJR3
7jj3JbJXLDL+n/hv8yvi/Txo1yZ4IKiW+dBPYJAmgnUdVWVtvZZ1qAx
sqfSINGtADVidWmrmhDWo4BvwhEv2OZzunx+ZUmftbzm+TOWdKsfBRxyhSo8wx0T
NlEE56SEtRBs/Nw63+KhEYPYBKCcgPP50uqafs2oYiTGIeYbWG3+9hIxz3u4rG31
(..)
HyFowjX5YnlGBoaydPLKvTFExskl9R3kMyU4k6+pW6F5n3fI7CDeFzsIZlD+o8Zr
e60NBekf0VN7goUv1XwZM3kb6/QdlApPh0igARaf8xK0hEzX9xIMjeoZySYvT0I9
3ytLeXKsVx42G+zJ/0FEnhKHrVxAIoP1sQJzsnhj/305LYhCf0N3xXaUKENx
/+cXc3r1p9rsIAewVa52wpaI8b6uIs97rB/iO7H/cm+Rj+9SYGSF3xGF6PGU1uhn
kqyb3WQJ4DtVsJ+ytO+vSmfNii1yEA8+/cWkQHHl5fCay9mKbbh71ZNR4qN0fupc
(..)
lXBw0ySdDByIUkaoLanGn3g3LYHitoXBD8Va4H//mWoMUsXQo7kBwWndwd+rFPy/
vqrbjPyelwf/i4+Y3h8Ek5yFlg711M1w0sYmDI1S0UzHmMGlJ2Nm1QHiOD1k9DQ5
tUrLWomrYJA6YgqSAnNglpGj3RfusVvLtPE0Z6Y3gzA7MB8wBwYFKw4DAhoEFO/x
zHZ0tJ4t1pGZUXGgqg3wCBKaBBQPqMtTh66bqURtgQkWjJ9JqGpQSQICB9A=
```

Chapter 3 Monitoring

3.1 Introduction

In our routers, IPsec monitoring is done once the SPD elements have been configured.

The difference, in terms of configuration, is that we will be listing parameters instead of changing them. As a result, any changes we make will only be temporary in nature (since they will no longer have any effect after we have re-booted the device).

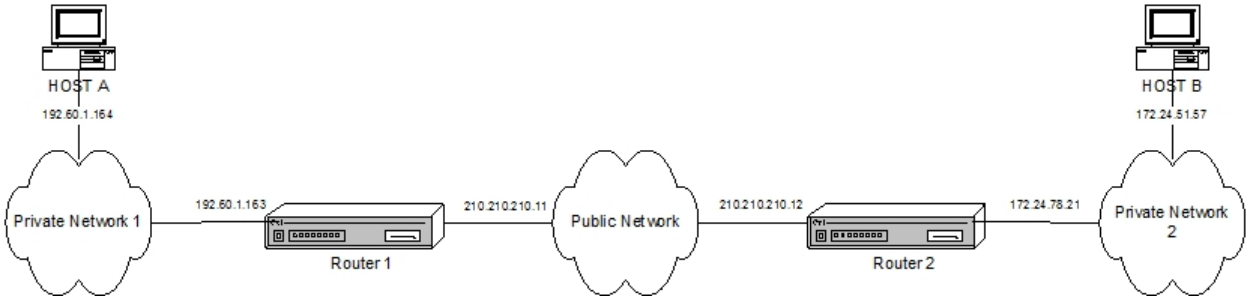
As explained in the introduction to this manual, security associations (SAs) are the secure connections created after consulting the SPD, and contain the necessary security information (authentication and encryption keys) to process a packet. Therefore, when we create an SA, we establish a secure connection for transmitting data between two tunnel endpoints.

There are two types of SA: phase I or ISAKMP SAs and phase II SAs. Phase II SAs can be dynamic or manual. There is a clear difference between the dynamic and ISAKMP SAs and the manual SAs, and this is something that should be taken into account. Manual SAs are permanent connections (i.e., the connection is set up between the tunnel endpoints when the manual templates are configured). Dynamic and ISAKMP SAs, being dynamic, only appear when we are using the connection between the tunnels endpoints (i.e., when the tunnel is set up).

The monitoring system lists the connections configured previously: the phase I or ISAKMP SAs, and the phase II or dynamic and manual SAs. It also allows us to eliminate connections, among other options.

In the following sections we will first describe the steps for accessing the monitoring system. Then we will explain each of the available commands in detail. Finally, we will outline some likely problems and solutions that may be encountered in IPsec negotiations.

The examples shown in each monitoring command are based on the following scenario.



3.2 IPsec monitoring

3.2.1 Initial monitoring

This section walks you through the steps required to access IPsec monitoring on our router. You need to enter the following commands to access the monitoring environment:

```
*p 3
Console operator
+protocol ip

-- IP protocol monitor --

IP+ipsec

-- IPSec protocol monitor --

IPSec+
```

The following commands are available within the IPsec protocol monitoring environment:

Command	Operation
? (HELP)	Lists the commands or their available options.
ADDRESS-TO-BAN	Enters the IPv4 addresses banned from using the protocol.
BITRATE	Displays the encapsulation/decapsulation rate in real time.

CERT	Enters the certificate monitoring menu.
CLEAR	Clears the cache memory and the security associations (*SAs*).
FAULT-TOLERANT	Enters the fault tolerant IPsec recovery monitoring menu.
FILTER-BY-HOST	Monitors only those events of a specific host name.
FILTER-DPD	Includes dead peer detection (*DPD*) events in the log.
GDOI GROUP	Enters the Group Domain Of Interpretation group monitoring menu.
HARDWARE	Functions relating to encryption cards (hardware encryption).
HOSTNAME-TO-BAN	Enters the host names that are banned from using the protocol.
LIST	Lists protocol elements.
MONITOR-LEVEL	Sets the monitoring level.
NO	Negates a command or restores its default value.
SHUTDOWN	Performs an orderly shutdown of all open connections and disables the protocol.
STOP-ON-MESSAGE	Stops the negotiation process message log at a specific message.
TEMPLATE-BITRATE	Displays the encapsulation/decapsulation rate in real time on a template.
EXIT	Exits the IPsec monitoring menu.

3.2.2 Monitoring commands

3.2.2.1 address-to-ban

Allows you to prevent certain IP addresses from using IPsec by specifying an IP address range that cannot include a security association (SA) source or destination address.

“ADDRESS-TO-BAN [IP add][mask]”

Prevents an IP address range defined by [IP add][mask] from using IPsec.

Example:

```
IPSec+address-to-ban 210.210.210.0 255.255.255.0
IPSec+
```

When you want to access information on blocked addresses, you should use the *LIST BANNED* and *NO ADDRESS-TO-BAN* commands to unblock the addresses.

3.2.2.2 bitrate

Allows you to monitor the protocol's packet encapsulation/decapsulation rate in real time. When there is a significant rate change, a new entry will appear in the table. You can halt monitoring at any time by pressing any key.

Example:

```
IPSec+bitrate
 Enc rate (bps/pps)  Dec rate (bps/pps)
 ------------------  ------------------
        480/    1         480/    1 (15:29:24)
       1136/    3         808/    2 (15:29:35)
       1456/    4        1808/    5 (15:29:36)
        480/    1         480/    1
```

3.2.2.3 cert

Lets you access the certificate monitoring menu. This is explained in the section on *Certificate monitoring commands* on page 170.

Example:

```
IPSec+cert

-- Cert user monitoring --
CERTIFICATES monit+
```

3.2.2.4 clear

You get the following subcommands when you select *CLEAR*.

Command	Function
COUNTERS	Clears counters from the encryption queue and from the SAs used.
SA	Cuts established SA connections.
STATISTICS	Clears the protocol's statistics.

3.2.2.4.1 clear counters

Clears counters from the encryption queue and from the SAs used. Use the *LIST ADVANCED* command to view the counter content.

Example:

```
IPSec+clear counters
  All counters have been reset.
IPSec+
```

3.2.2.4.2 clear sa

We can use this command to cut the connection established between the two tunnel endpoints. The type of interruption will depend on the type of SA we have.

There is little point in removing a manual SA because, as we saw above, the connection is permanent and cannot be cut. We can, however, clear the dynamic and ISAKMP SAs.

Command	Operation
ALL	Removes all dynamic and ISAKMP SAs.
HOSTNAME-FILTER	Removes those SAs involving a specific device.
IN	Removes incoming dynamic SAs.
NEGOTIATION	Removes ISAKMP or phase I SAs.
OUT	Removes outgoing dynamic SAs.

“CLEAR SA ALL”

Clears all ISAKMP and dynamic SAs.

Example:

```
IPSec+clear sa all
 Clearing IPSec Connections...  Done
IPSec+
```

“CLEAR SA HOSTNAME-FILTER [hostname]”

Deletes all dynamic and ISAKMP SAs involving a specific device (determined by [hostname]). You can use the asterisk character (*) to include all *hostnames* beginning with the same characters. In the following example, all SAs involving devices whose *hostname* begins with “HOST_”, are removed.

Example:

```
IPSec+clear sa hostname-filter HOST_*
 HOST_*-->70.70.70.2
 Connection 2 cleared
 Connection 3 cleared
 Connection 1 cleared
IPSec+
```

You will get the following options when you select any of the commands between *CLEAR SA IN*, *CLEAR SA NEGOTIATION* and *CLEAR SA OUT*:

Command	Function
ADDRESS-FILTER	Removes all SAs containing an IPv4 source or destination address within in a specified range.
ALL	Removes all SAs of the selected type.
CONNECTION	Removes all SAs with a specified ID number.
HOSTNAME-FILTER	Removes all SAs involving a specific device.
IKEV1	Removes all SAs negotiated with IKEv1 protocol. Available only on "clear sa negotiation" command
IKEV2	Removes all SAs negotiated with IKEv2 protocol. Available only on "clear sa negotiation" command

“CLEAR SA IN/NEGOTIATION/OUT ADDRESS-FILTER [ip add][mask]”

Clears all SAs (incoming dynamic/ISAKMP/outgoing dynamic) containing an IPv4 source or destination address that fall within the range specified by [ip add][mask].

Example:

```
IPSec+clear sa negotiation address-filter 210.210.210.12 255.255.255.255
 Connection 1 cleared
IPSec+
```

“CLEAR SA IN/NEGOTIATION/OUT ALL”

Clears the selected type of SA (incoming dynamic/ISAKMP/outgoing dynamic).

Example:

```
IPSec+clear sa negotiation all
 All IPSec connections cleared
IPSec+
```

“CLEAR SA IN/NEGOTIATION/OUT CONNECTION [id]”

The [id] field is the SA identification number. All dynamic and ISAKMP SAs with the specified [id] number are removed.

Example:

```
IPSec+clear sa negotiation connection 1
 Connection 1 cleared
IPSec+
```

“CLEAR SA IN/NEGOTIATION/OUT HOSTNAME-FILTER [hostname]”

Clears all SAs (incoming dynamic/ISAKMP/outgoing dynamic) involving a specific device, determined by [hostname].

Example:

```
IPSec+clear sa negotiation hostname-filter HOST_H
 HOST_H-->70.70.70.2
 Connection 1 cleared
IPSec+
```

“CLEAR SA NEGOTIATION IKEV1”

Clears all SAs (incoming dynamic/ISAKMP/outgoing dynamic) negotiated using IKEv1 protocol.

Example:

```
IPSec+clear sa negotiation ikev1
 Connection 1 cleared
IPSec+
```

“CLEAR SA NEGOTIATION IKEV2”

Clears all SAs (incoming dynamic/ISAKMP/outgoing dynamic) negotiated using IKEv2 protocol.

Example:

```
IPSec+clear sa negotiation ikev2
 Connection 2 cleared
IPSec+
```

Command history:

Release	Modification
11.01.10	The IKEV1 and IKEV2 options were introduced on CLEAR SA NEGOTIATION command as of version 11.01.10.

3.2.2.4.3 clear statistics

Deletes the protocol's statistics. You can use the *LIST STATISTICS* command to show these statistics.

Example:

```
IPSec+clear statistics
  All IPSec statistics have been reset.
IPSec+
```

3.2.2.5 fault-tolerant

Lets you access the fault tolerant IPsec recovery monitoring menu. This will be explained in the section on *Certificate monitoring commands* on page 170.

Example:

```
IPSec+fault-tolerant

-- Fault tolerant IPSec recovery monitor --
IPSecFT monitor+
```

3.2.2.6 filter-by-host

FILTER-BY-HOST allows you to limit protocol monitoring to only those events involving a specific device, identified as *hostname*.

“FILTER-BY-HOST [hostname]”

Limits monitoring to events relating to the device whose host name is [hostname].

Example:

```
IPSec+filter-by-host HOST_H
 Filter activated with hostname HOST_H
IPSec+
```

Use the *LIST NEGOTIATION FILTER* command to view the [hostname] defined in the filter. Use the *NO FILTER-BY-HOST* command to disable it.

3.2.2.7 filter-dpd

Enables the filter showing dead peer detection *(DPD)* protocol events and logs.

Example:

```
IPSec+filter-dpd
IPSec+
```

You can use the *LIST NEGOTIATION FILTER* command to check whether the filter is enabled. Use *NO FILTER-DPD* to disable it.

3.2.2.8 GDOI group

Allows you to enter the Group Domain Of Interpretation (GDOI) group monitoring menu. This will be explained in the section on *GDOI group monitoring commands* on page 179.

Example:

```
IPSec+gdoi group 123

-- GDOI group monitor --
IPSec GDOI monitor+
```

3.2.2.9 hardware

Command	Function
enable	Enables an encryption card.
list	Lists the status of hardware used by IPsec.
test	Analyzes whether encryption hardware is present.

“HARDWARE ENABLE {cf1531/fsl-dpaa-sec/fsl-sec/generic-driver/mpc8272/mpc85xx/ts422-ts727}”

Enables the selected hardware component, if available, to accelerate the encryption process.

Example:

```
IPSec+hardware enable ?
  cf1531            CF1531
  fsl-dpaa-sec      FSL-DPAA-SEC
  fsl-sec           FSL-SEC
  generic-driver    GENERIC DRIVER
  mpc8272           MPC8272
  mpc85xx           MPC85XX
  ts422-ts727       TS422-TS727
IPSec+hardware enable fsl-dpaa-sec
IPSec+
```

Command history:

Release	Modification
11.01.00	The *FSL-DPAA-SEC* option was introduced as of version 11.01.00.

The *HARDWARE LIST* command shows whether the encryption card is enabled. Use the *NO HARDWARE ENABLE* command to disable it.

“HARDWARE LIST”

Lists the status of the hardware used to accelerate the IPsec encryption process.

Example:

```
IPSec+hardware list

Hardware: FSL-SEC-5.0 Revision: 0x500, block 0xA12
    Status: OK. Access enabled.
    Free jobs 64  Busy jobs 0
    Busy jobs watermark 1 ISR watermark 0 completed watermark 1
    Interrupts: 562 Spurious: 0
       Job ring 0  status 0
           in: 1 out:1 in avail:64 out used:0
        DES encode:512  DES decode:8
        AES encode:1  AES decode:1
        HASH:21
        RSA:27
        RNG:2
IPSec+
```

“HARDWARE TEST”

Tests the encryption cards that are enabled on the device.

Example:

```
IPSec+hardware test ?
  <0s..52w1d>    Time value
IPSec+hardware test 5s
IPSec+
```

3.2.2.10 hostname-to-ban

Allows you to block a device from using IPsec by specifying a host name that must not form part of any security association.

“HOSTNAME-TO-BAN [hostname]”

Prevents the router from establishing an IPsec tunnel with a device whose *hostname* is [hostname].

Example:

```
IPSec+hostname-to-ban HOST_H
IPSec+
```

Use the *LIST BANNED* command to access information on blocked devices. Use the *NO HOSTNAME-TO-BAN* command to unblock the devices.

3.2.2.11 list

Shows protocol monitoring information. You can use the following commands:

Command	Function
ACCESS-LISTS	Displays information on access lists.
ADDRESS-FILTER	Provides protocol information on IP addresses within a certain range.
ADVANCED	Displays the content of the counters used in the SA and encryption queue.
BANNED	Lists addresses or devices that cannot use IPsec.
CERTIFICATE_NUMBER	Certificate identified by the number assigned during IKE negotiation.
HOSTNAME-FILTER	Provides protocol information on a device identified by its *hostname*.
NEGOTIATION	IKE negotiation log (*Internet Key Exchange*).
NOTIFICATION	IKE negotiation notification messages.
SA	Security associations.
STATISTICS	IPsec protocol operating statistics.

3.2.2.11.1 list access-lists

The following options become available when you enter this command:

Command	Function
address-filter	Displays all the protocol information relating to the specified IP address range.
all	Displays all access list information (cache and entries).
Cache	Displays cache information from access lists associated with IPsec.
entries	Displays defined entries from the access lists associated with IPsec.

“LIST ACCESS-LIST ADDRESS-FILTER [IP add][mask]”

The information displayed is limited to cases involving an IP address in the range [IP add][mask].

“LIST ACCESS-LIST ALL”

Displays all available information on the access lists assigned to IPsec, formed from cache and the list entries.

“LIST ACCESS-LIST CACHE”

Displays all the cache data for the access lists assigned to IPsec.

“LIST ACCESS-LIST ENTRIES”

Lists all the entries defined in the access lists assigned to IPsec.

3.2.2.11.2 list address-filter

“LIST ADDRESS-FILTER [IP add][mask]”

Selects and displays all IPsec monitoring information relating to the IP address range [IP add][mask].

Example:

```
IPSec+list address-filter 10.10.10.1 255.255.255.255

SA OUT
   SA 3 SPI=0x725d86d5
   SA UP, ESP-DES ESP-MD5     SRC=10.10.10.1 DES=10.10.10.2
   LifeTime:1h0m0s (0h56m7s)
   encode pkts:2 (err:0)
   Path MTU: 1446, number of path MTU sent 0, number of path MTU received 0
        SRC=1.1.1.1/32 DES=2.2.2.2/32 ethernet0/0


SA IN
   SA 2 SPI=0x8426bab4
   SA UP, ESP-DES ESP-MD5     SRC=10.10.10.2 DES=10.10.10.1
   LifeTime:1h0m0s (0h56m7s)
   decode pkts:2 (err:0)
   DPD: idle for 51(60) seconds
```

```
       SRC=1.1.1.1/32 DES=2.2.2.2/32 ethernet0/0


SA NEGOTIATION
  SA 1 (i_cookie=0x172ba5e581af8c2b r_cookie=0x2f5b71cea58696b7)
  Inic=10.10.10.1  Resp=10.10.10.2
  SRC=10.10.10.1 DES=10.10.10.2 STATE=5
   LifeTime:1h0m0s (0h56m7s)
   ClientSRC=1.1.1.1/32 ClientDES=2.2.2.2/32 Rule=0 Ifc=ethernet0/0
   DPD ENABLED : idle for 60(60) seconds, ignored 0(3) DPD packets, waiting for out-traffic
   IKE fragmentation
   ISAKMP_SA available, STATE=ESTABLISH
      ISAKMP_NEGII id 0x378218c9, (0x8426bab4/0x725d86d5)
        SRC=1.1.1.1/32 DES=2.2.2.2/32 ethernet0/0
          LifeTime:1h0m0s (0h56m7s)
          encode pkts:2 (err:0), decode pkts:2 (err:0)


Extended Access List 100, assigned to IPSec

 ACCESS LIST CACHE. Hits = 0, Miss = 1
 Cache size: 32 entries, Promotion zone: 6 entries

 ACCESS LIST ENTRIES
IPSec+
```

3.2.2.11.3 list advanced

Lets you read the encryption queue and SA counter values used. It also tells you whether the protocol has been disabled with the *SHUTDOWN* command.

Example:

```
IPSec+list advanced
  Cipher Queue Size:             50
  Cipher Queue Water Mark:       2  reached 0h4m27s ago.
  Current Queue Level:           0
  Max SA simultaneous:           2  reached 0h4m27s ago.
  Current number of SA Out:      1
  Current number of SA In:       1
  Max tunnel supported:          400
  Access list hash size (slots): 37
  Number of keys in hash:        48
  Number of free keys:           1
  Fields checked by hash:        ia_src ia_dst selector1 selector2 selector3
  Hash usage (absolute):         2 (ipsec) + 635 (other) + 48 (fail) = 685 (total)
  Hash usage (relative):         0% (ipsec) + 92% (other) + 7% (fail)
  Hash usage (moving average):   0.1% (ipsec) + 96.3% (other) + 3.5% (fail)
  Hash distribution:
    [min ... max] -> number of slots with n keys, min <= n <= max
        [0 ... 0] -> 12
        [1 ... 1] -> 11
        [2 ... 3] -> 12
        [4 ... 7] -> 2
  Previous exponentiation level 1 (max 1)

  IPSec active
IPSec+
```

Use the *CLEAR COUNTERS* command to reset the counter values.

3.2.2.11.4 list banned

Lists an IP address range and any host names that are not allowed to form part of an IPsec tunnel because they have been banned via the *ADDRESS-TO-BAN* or *HOSTNAME-TO-BAN* commands.

Example:

```
IPSec+list banned
```

```
Banned addresses:
210.210.210.0

Banned hostnames:
HOST_H

IPSec+
```

3.2.2.11.5 list certificate_number

“LIST CERTIFICATE_NUMBER [id]”

Displays information about a particular certificate identified by the number [id] assigned in the IKE negotiation.

3.2.2.11.6 list hostname-filter

“LIST HOSTNAME-FILTER [hostname]”

Selects and displays all monitoring information relating to the device whose host name is [hostname]. As with other IPsec commands where you have to enter a [hostname], you can use the asterisk character (*) to select all the devices whose host name begins with the characters preceding the asterisk.

3.2.2.11.7 list negotiation

Nested within this command, and related to the IKE negotiation log, are the following commands:

Command	Function
ADDRESS-FILTER	Only reports negotiations involving a specific device, determined by its IP address.
ALL	Displays the negotiation in full.
BETWEEN	Displays the IKE negotiation between two specific devices.
FILTER	Specifies which filters are enabled.
HOSTNAME-FILTER	Only reports negotiations involving a specific device, determined by its *hostname*.
ORDER	Displays the negotiation in order of conversations among peers.

“LIST NEGOTIATION ADDRESS-FILTER [IP add]”

Only shows negotiations for entries involving the device whose IPv4 address is [IP add].

“LIST NEGOTIATION ALL”

Shows the entire IKE negotiation that has not yet been consulted.

Example:

```
IPSec+list negotiation all

10.10.10.1 10.10.10.2: (09:04:39)
              (* 09:--------- Creating ISAKMP SA id ---------)(# 2035380069(0x79516f65))
              (* 09:--------- Creating ISAKMP SA id ---------)(# -1155452362(0xbb213236))
  (HDR 79516f65)
      (HDR hash)
      (del isakmp #1)(* 172ba5e581af8c2b2f5b71cea58696b7)
  (HDR bb213236)
      (HDR hash)
      (HDR sa)
      (prop 1 esp #1)(# 461703945(0x1b850b09))
      (trans 1 id=des)
              (encap tunnel)
              (life sec)
              (duration 3600)
              (auth alg md5)
      (HDR nonce)(# 24(0x18))
      (id addr4 prot=0 port=0)(# 1.1.1.1)
      (id addr4 prot=0 port=0)(# 2.2.2.2)
10.10.10.2 10.10.10.1: (HDR bb213236)
      (HDR hash)
      (HDR sa)
      (prop 1 esp #1)(# -243871082(0xf176d296))
```

```
      (trans 1 id=des)
              (encap tunnel)
              (life sec)
              (duration 3600)
              (auth alg md5)
      (HDR nonce)(# 24(0x18))
      (id addr4 prot=0 port=0)(# 1.1.1.1)
      (id addr4 prot=0 port=0)(# 2.2.2.2)
10.10.10.1 10.10.10.2:
              (* 01:--------- Matching template ---------)(# 2(0x2))
  (HDR bb213236)
      (HDR hash)
              (* 11:--------- Creating SA IN ---------)(# 461703945(0x1b850b09))
              (* 12:--------- Creating SA OUT ---------)(# -243871082(0xf176d296))
              (* 40:------!!! CONNECTED !!!------)
              (* 09:--------- Creating ISAKMP SA id ---------)(# -2093442493(0x83389a43))
10.10.10.2 10.10.10.1: (HDR 83389a43)
      (HDR hash)
      (del esp #1)(* 8426bab4)
10.10.10.1 10.10.10.2: (09:04:40)
              (* 14:--------- Deleting SA IN ---------)(# -2077836620(0x8426bab4))
              (* 09:--------- Creating ISAKMP SA id ---------)(# 787648381(0x2ef28f7d))
              (* 15:--------- Deleting SA OUT ---------)(# 1918731989(0x725d86d5))
  (HDR 2ef28f7d)
      (HDR hash)
      (del esp #1)(* 725d86d5)
  (09:04:43)
              (* 09:--------- Creating ISAKMP SA id ---------)(# 1173201016(0x45eda078))
              (* 14:--------- Deleting SA IN ---------)(# 461703945(0x1b850b09))
              (* 09:--------- Creating ISAKMP SA id ---------)(# -766098472(0xd25643d8))
              (* 15:--------- Deleting SA OUT ---------)(# -243871082(0xf176d296))
  (HDR 45eda078)
      (HDR hash)
      (del esp #1)(* 1b850b09)
  (HDR d25643d8)
      (HDR hash)
      (del esp #1)(* f176d296)
  (09:04:47)(* 07:--------- Deleting ISAKMP NEG ---------)(# 1(0x1))
  (09:05:01)
              (* 36:--------- Local Starting Neg ---------)
              (* 06:--------- Creating ISAKMP NEG ---------)(# 6(0x6))
  (HDR 0)
      (HDR sa)
      (prop 1 isakmp #1)
      (trans 1 id=1)
              (encrypt des)
              (hash md5)
              (grp desc 1)
              (auth presh)
              (life sec)
              (duration 3600)
      (vendor 13)(*  lJ/M  GeBintec)
      (vendor attrcfg)
      (vendor xauth)
      (vendor ikefrag)
      (vendor natt)(* nat-t-rfc)
      (vendor dpd)
10.10.10.2 10.10.10.1: (HDR 0)
      (HDR sa)
      (prop 1 isakmp #1)
      (trans 1 id=1)
              (encrypt des)
              (hash md5)
              (grp desc 1)
              (auth presh)
              (life sec)
              (duration 3600)
```

```
      (vendor 13)(* edbc9a71ce06efadb354656c6461740d)
      (vendor attrcfg)
      (vendor xauth)
      (vendor ikefrag)
      (vendor natt)(* nat-t-rfc)
      (vendor dpd)
10.10.10.1 10.10.10.2:
              (* 01:--------- Matching template ---------)(# 1(0x1))
  (HDR 0)
      (HDR keyx)
      (HDR nonce)(# 24(0x18))
      (HDR natd)(* 34611df8bf992fb7c5daf89f06095c7c)
      (HDR natd)(* 8276dfc47c7d841133f5370418eb37a6)
10.10.10.2 10.10.10.1: (HDR 0)
      (HDR keyx)
      (HDR nonce)(# 24(0x18))
      (vendor 13)(* edbc9a71ce06efadb354656c6461740d)
      (vendor attrcfg)
      (vendor xauth)
      (vendor ikefrag)
      (vendor natt)(* nat-t-rfc)
      (vendor dpd)
      (HDR natd)(* 8276dfc47c7d841133f5370418eb37a6)
      (HDR natd)(* 34611df8bf992fb7c5daf89f06095c7c)
10.10.10.1 10.10.10.2:
              (* 08:--------- Creating ISAKMP SA ---------)
  (HDR 0)
      (id addr4 prot=17 port=500)(# 10.10.10.1)
      (HDR hash)
10.10.10.2 10.10.10.1: (HDR 0)
      (id addr4 prot=17 port=500)(# 10.10.10.2)
      (HDR hash)
10.10.10.1 10.10.10.2:
              (* 09:--------- Creating ISAKMP SA id ---------)(# -825652044(0xcec98cb4))
  (HDR cec98cb4)
      (HDR hash)
      (HDR sa)
      (prop 1 esp #1)(# -436391738(0xe5fd30c6))
      (trans 1 id=des)
              (encap tunnel)
              (life sec)
              (duration 3600)
              (auth alg md5)
      (HDR nonce)(# 24(0x18))
      (id addr4 prot=0 port=0)(# 1.1.1.1)
      (id addr4 prot=0 port=0)(# 2.2.2.2)
10.10.10.2 10.10.10.1: (HDR cec98cb4)
      (HDR hash)
      (HDR sa)
      (prop 1 esp #1)(# 1280635136(0x4c54f100))
      (trans 1 id=des)
              (encap tunnel)
              (life sec)
              (duration 3600)
              (auth alg md5)
      (HDR nonce)(# 24(0x18))
      (id addr4 prot=0 port=0)(# 1.1.1.1)
      (id addr4 prot=0 port=0)(# 2.2.2.2)
10.10.10.1 10.10.10.2:
              (* 01:--------- Matching template ---------)(# 2(0x2))
  (HDR cec98cb4)
      (HDR hash)
              (* 11:--------- Creating SA IN ---------)(# -436391738(0xe5fd30c6))
              (* 12:--------- Creating SA OUT ---------)(# 1280635136(0x4c54f100))
              (* 40:------!!! CONNECTED !!!------)
0.0.0.0: (09:05:14)
IPSec+
```

“LIST NEGOTIATION BETWEEN [IP1 add][IP2 add]”

Selects and displays negotiations between devices whose IPv4 addresses are [IP1 add] and [IP2 add].

“LIST NEGOTIATION FILTER”

Shows which of the following filters are enabled: *negotiation filter*, *event filter* and *DPD filter*. These are enabled through the *FILTER-BY-HOST* and *FILTER-DPD* commands, or using the *EVENT ADDRESS-FILTER* command from the IPsec configuration menu; the latter of these commands creates an *event filter* with a range of IP addresses to monitor. These filters affect the output of the *BITRATE*, *LIST NEGOTIATION* and *LIST STATISTICS* commands.

Example:

```
IPSec+list negotiation filter

Negotiation filter:  Hostname:HOST_* ,  Address:0.0.0.0

Event filter: Hostname:HOST_* ,  Address:255.255.255.255

DPD filter

IPSec+
```

“LIST NEGOTIATION HOSTNAME-FILTER [hostname]”

Filters out all entries except those involving the device identified by [hostname].

3.2.2.11.8 list notification

Displays IKE negotiation notification messages: failed negotiation proposals, incompatible, deleted SAs, etc.

Example:

```
IPSec+list notification



  (09:10:44)
IPSec+
```

3.2.2.11.9 list sa

Allows us to view all the SAs and see whether the connections are active.

When we list the SAs, manual SAs will always be shown because they are permanent connections. Since the dynamic and ISAKMP SAs are dynamic, they will only be shown if we are using the connection between the tunnel ends (i.e., if we are transmitting data).

Command	Operation
IN	Lists incoming dynamic and manual SAs.
NEGOTIATION	Lists ISAKMP or phase I SAs.
OUT	Lists outgoing dynamic and manual SAs.

Each of these commands is also associated with a final set of options:

Command	Function
address-filter	Only lists the SAs that contain an IP address within a certain range.
all	Displays information on all SAs of the selected type.
connection	(*For negotiation only*) Only lists the ISAKMP SA specified by the identifier.
hostname	(*For negotiation only*) Lists the ISAKMP SAs involving a specific device.
ikev1	Lists IKEv1 elements.
ikev2	Lists IKEv2 elements.

“LIST SA IN/NEGOTIATION/OUT ADDRESS-FILTER [IP add][mask]”

Filters out all SAs (incoming/ISAKMP/outgoing) except those that contain a source or destination address within the range defined by [IP add][mask].

Example:

```
IPSec+list sa negotiation address-filter 10.10.10.1 255.255.255.255
```

```
SA NEGOTIATION
  SA 6 (i_cookie=0xa65b782f4d068855 r_cookie=0xdc8cac71ce06ef94)
  Inic=10.10.10.1  Resp=10.10.10.2
  SRC=10.10.10.1 DES=10.10.10.2 STATE=5
   LifeTime:1h0m0s (0h53m3s)
   ClientSRC=1.1.1.1/32 ClientDES=2.2.2.2/32 Rule=0 Ifc=ethernet0/0
   DPD ENABLED : idle for 60(60) seconds, ignored 0(3) DPD packets, waiting for out-traffic
   IKE fragmentation
   ISAKMP_SA available, STATE=ESTABLISH
      ISAKMP_NEGII id 0xcec98cb4, (0xe5fd30c6/0x4c54f100)
        SRC=1.1.1.1/32 DES=2.2.2.2/32 ethernet0/0
          LifeTime:1h0m0s (0h53m3s)
          encode pkts:2 (err:0), decode pkts:2 (err:0)

IPSec+
```

“LIST SA IN/NEGOTIATION/OUT ALL”

Lists all the SAs of the selected type that are currently active (incoming/ISAKMP/IKEv2/outgoing). In the case of incoming and outgoing SAs, it will list the manual SAs and the active dynamic SAs.

Example:

```
IPSec+list sa in all

SA IN
   SA 7 SPI=0xe5fd30c6
   SA UP, ESP-DES ESP-MD5     SRC=10.10.10.2 DES=10.10.10.1
   LifeTime:1h0m0s (0h52m41s)
   Direct-decoded-fwd: nexthop 20.20.20.2, ifc ethernet0/1
   decode pkts:2 (err:0) (FVRF err:0 - IVRF err:0)
   DPD: idle for 17(60) seconds
        SRC=1.1.1.1/32 DES=2.2.2.2/32 ethernet0/0

IPSec+
```

The IKEv2 information displayed is as follows:

```
IPSec+list sa negotiation all

SA NEGOTIATION
  SA 327 [IKEv2] (i_cookie=0xf0fd1c34fb2bbac3 r_cookie=0xe17a6c2fef5a37fd)
  Inic=192.168.212.204  Resp=192.168.212.203
  SRC=192.168.212.203 DES=192.168.212.204 STATE=6
   LifeTime:1h0m0s (0h26m52s)
   ClientSRC=192.168.212.203/32 ClientDES=192.168.212.204/32 Rule=0 Ifc=ethernet0/0 GRE
   DPD ENABLED : idle for 60(60) seconds, ignored 0(3) DPD packets, waiting for out-traffic
   ISAKMP_SA available, STATE=ESTABLISH
      CHILD_SA id 0xb7559a17, (0x7598c4be/0x5a6a1514)
        SRC=192.168.212.203/32 DES=192.168.212.204/32 GRE  ethernet0/0
          LifeTime:1h0m0s (0h26m55s)
          encode pkts:0 (err:0), decode pkts:0 (err:0)
```

“LIST SA IN/NEGOTIATION/OUT IKEv1 | IKEv2”

Filters out all SAs (incoming/ISAKMP/outgoing) except for IKEv1 or IKEv2.

“LIST SA NEGOTIATION HOSTNAME [hostname]”

Shows active ISAKMP SAs built between the device and the IPsec tunnel peer identified by [hostname].

“LIST SA NEGOTIATION CONNECTION <SA ID>”

Shows the ISAKMP SA specified by the identifier.

Command history:

Release	Modification
11.00.05	As of version 11.00.05, the output of the *LIST SA IN* and *LIST SA OUT* commands only shows the statistic that applies (i.e., decoded packets for SA IN, and encoded packets for

Release	Modification
	SA OUT) rather than both statistics (encoded and decoded packets). They also show information related to the *DIRECT-DECODED-FWD* and *DIRECT-ENCODED-FWD* commands.
11.01.00	As of version 11.01.00, the output of the *LIST SA IN* and *LIST SA OUT* commands only shows the statistic that applies (i.e., decoded packets for SA IN, and encoded packets for SA OUT) rather than both statistics (encoded and decoded packets). They also show information related the *DIRECT-DECODED-FWD* and *DIRECT-ENCODED-FWD* commands.
11.00.07	As of version 11.00.07, the output of the *LIST SA NEGOTIATION* , *LIST SA IN* and *LIST SA OUT* commands shows IKEv2 SAs.
11.01.02	As of version 11.01.02, the output of the *LIST SA NEGOTIATION* , *LIST SA IN* and *LIST SA OUT* commands shows IKEv2 SAs.
11.01.04	The *IKEV1*, *IKEV2* and *CONNECTION* options were introduced as of version 11.01.04.
11.01.04	As of version 11.01.04, the output of the *LIST SA NEGOTIATION* command shows the routes received via IKEv2 Configuration Payload.
11.01.08	As of version 11.01.08, the output of the *LIST SA NEGOTIATION* command shows Transport encapsulation information.
11.01.08	As of version 11.01.08, the output of the *LIST SA IN* and the *LIST SA OUT* commands shows VRF error counter information.

3.2.2.11.10 list statistics

Lists the IPsec protocol performance statistics.

Example:

```
IPSec+list statistics

----ESP/AH Statistics:----

Input Stats
-----------
  Frames ok    4
  Frames error 0
  ---> Out-of-Order frames      0
  ---> Unknown payload protocol 0
  ---> Internal errors          0
  Frames/sec 0 (max 1)
  kbits/sec 0 (max 0)
Output Stats
------------
  Frames ok    4
  Unknown authentication algorithm 0
  Frames/sec 0 (max 1)
  kbits/sec 0 (max 0)


----IPSEC Forwarding Statistics:----

Sa in not found                0
Invalid spi notifications sent 0
Sa out Template not found      0
Sa out not found(only manual)  0

----IKE Statistics:----
ISAKMP not found               0
Invalid ISAKMP notif. sent     0
Negotiation phase I            1
Negotiation phase II           1
Check Hash Error phase I       0
Check Hash Error phase II      0
Drops Collision IKE messages   0
Drops Waiting IKE Processing   0
```

```
   Cypher queue empty:          0
   Job queue empty:             0

   Number of open connections not notified during last connevent-period: 0


IPSec+
```

Use the *CLEAR STATISTICS* command to restart the corresponding variables.

3.2.2.11.11 list tunnel-protection

Lists the tunnel protection template active configuration.

Example:

```
Router1 IPSec+list tunnel-protection
 TEMPLATE 2:
   Protecting interface tnip2 (tunnel mode IPSEC IP)
   Source-address 10.10.10.203
   Mapped-to-ifc tnip2
 TEMPLATE 4:
   Protecting interface tnip1 (tunnel mode IPSEC IP)
   Source-address 192.168.0.1
   Mapped-to-ifc tnip1
Router1 IPSec+
```

Command history:

Release	Modification
11.01.04	This monitoring command was introduced as of version 11.01.04.

3.2.2.12 monitor-level

The **verbose** option in the *MONITOR-LEVEL* command gives detailed monitoring information.

“MONITOR-LEVEL VERBOSE”

Enables detailed monitoring.

Example:

```
IPSec+monitor-level verbose
IPSec+
```

Use the *NO MONITOR-LEVEL VERBOSE* command to disable the *verbose* monitoring mode.

3.2.2.13 no

The sole purpose of this command is to perform the opposite action of the other protocol commands. The menu shown in **no** is a replica of the initial menu, but only contains those commands for which there is an opposite function.

Command	Function
ADDRESS-TO-BAN	Unblocks IPv4 addresses banned from IPsec.
FILTER-BY-HOST	Stops limiting monitoring to a specific device.
FILTER-DPD	Does not include DPD protocol events and logs.
HARDWARE	Using the **enable** option, disables a specific encryption card.
HOSTNAME-TO-BAN	Unblocks devices banned from IPsec.
MONITOR-LEVEL	Using the **verbose** option reduces the amount of detailed monitoring information shown.
SHUTDOWN	Enables IPsec.
STOP-ON-MESSAGE	Doesn't halt the negotiation log when a particular message occurs.

3.2.2.14 shutdown

Performs an orderly shutdown of all open IPsec connections and disables the protocol.

Example:

```
IPSec+shutdown

 Clearing IPSec Connections...100%   1 SANEGs cleared. Done
IPSec+
```

Warning

You will instantly lose access if you run this command when connecting via a remote console through an IPsec session. You won't be able to recover access until you reboot the device or until you run the *NO SHUTDOWN* command through an access mode that isn't protected by IPsec.

3.2.2.15 stop-on-message

Stops the negotiation message log when a particular message occurs. The options for this command are the numbers identifying the messages where the log can be stopped.

Command	Function
01	--------- Matching template ---------
02	--------- Matching SA NEG ---------
03	--------- Decryption error ---------
04	--------- Retransmission ---------
05	--------- Unable to initiate. Unknown destination ---------
06	--------- Creating ISAKMP NEG ---------
07	--------- Deleting ISAKMP NEG ---------
08	--------- Creating ISAKMP SA ---------
09	--------- Creating ISAKMP SA id ---------
10	--------- Unable to make ISAKMP SA ---------
11	--------- Creating SA IN ---------
12	--------- Creating SA OUT ---------
13	--------- Deleting AL ENTRY ---------
14	--------- Deleting SA IN ---------
15	--------- Deleting SA OUT ---------
16	--------- KeepAlive Deleting SA ---------
17	--------- Purgetime SA NEG ---------
18	--------- Purgetime SA NEG II ---------
19	--------- ISAKMP SA negotiating I ---------
20	--------- Negotiation on Phase I. Phase II not allowed ---------
21	--------- Invalid NegII or ISAKMP SA ---------
22	--------- Initiated Renegotiation Timer for SA OUT ---------
23	--------- Renegotiation Timer for SA OUT expired ---------
24	--------- Attempt to renegotiate delayed ---------
25	--------- Matching CRL ---------
26	--------- Invalid ID information ---------
27	--------- No Dir Info in SA ---------
28	--------- ISAKMP SA negotiating II ---------
29	--------- Negotiation from banned host stopped ---------
30	--------- DPD Deleting SAs ---------
31	--------- Max tunnel supported reached ---------
32	--------- Renegotiation using DNAT ---------
33	--------- Lifetime changed ---------
34	--------- Searching Backup Peer ---------
35	--------- Renegotiation Timer for SA NEG expired ---------
36	--------- Local Starting Neg ---------
37	--------- Remote Starting Neg ---------

38	--------- Local Starting BackUp Neg ---------
39	--------- Remote Starting BackUp Neg ---------
40	------!!! CONNECTED !!!------
100	*** Any Notify Message ***

In the example shown below, the negotiation log is interrupted when message 36 appears. After running the *STOP-ON-MESSAGE* command, all ISAKMP SAs are deleted in order to start a new negotiation, which will happen if there is IPsec traffic (the SAs are not permanent). We can list the negotiation to see how the log stopped when the message occurred.

Example:

```
IPSec+stop-on-message 36

Activated stop on message number 36
IPSec+list negotiation all



10.10.10.1 10.10.10.2: (09:26:29)
              (* 09:--------- Creating ISAKMP SA id ---------)(# -1060500311(0xc0ca0ca9))
              (* 14:--------- Deleting SA IN ---------)(# 391025728(0x174e9440))
              (* 09:--------- Creating ISAKMP SA id ---------)(# 459171182(0x1b5e656e))
              (* 09:--------- Creating ISAKMP SA id ---------)(# -83899945(0xfaffc9d7))
              (* 15:--------- Deleting SA OUT ---------)(# 495236952(0x1d84b758))
  (HDR c0ca0ca9)
      (HDR hash)
      (del esp #1)(* 174e9440)
  (HDR 1b5e656e)
      (HDR hash)
      (del esp #1)(* 1d84b758)
  (HDR faffc9d7)
      (HDR hash)
      (del isakmp #1)(* 7d85533ed2653d38a8c514d75784d1a8)
  (09:26:33)
              (* 17:--------- Purgetime SA NEG ---------)
  (09:26:34)(* 07:--------- Deleting ISAKMP NEG ---------)(# 20(0x14))
  (09:26:54)
              (* 36:--------- Local Starting Neg ---------)
0.0.0.0: (09:26:59)
**** REGISTRY STOPPED BY MESSAGE NUMBER 36 ****
IPSec+
```

3.2.2.16 template-bitrate

Allows you to monitor the protocol's packet encapsulation/decapsulation rate for the selected dynamic template, in real time. When there is a significant rate change, a new entry will appear in the table. You can halt monitoring at any time by pressing any key.

Example:

```
IPSec+template-bitrate 2
 TEMPLATE 2:
    Enc rate (bps/pps)  Dec rate (bps/pps)
    ------------------  ------------------
              0/    0           0/    0
IPSec+
```

Note

To enable real time monitoring of the packet encapsulation/decapsulation rate for a dynamic template, you must configure a *REPORT* related with *RATE* on this template. To do this, use the *TEMPLATE [ID] REPORT {TX-RATE-KBPS/TX-RATE-KBPS}* commands as described in the section on *Dynamic template parameters* on page 35.

```
IPSec+template-bitrate 2
 TEMPLATE 2:
    Enc rate (bps/pps)  Dec rate (bps/pps)
    ------------------  ------------------
```

```
        67200/  100       67200/  100 (09:41:37)
        64512/   96       64512/   96 (09:41:46)
        67200/  100       67200/  100 (09:41:47)
            0/    0           0/    0
IPSec+
```

Command history:

Release	Modification
11.00.05	This command was introduced as of version 11.00.05.
11.01.00	This command was introduced as of version 11.01.00.

3.2.3 Certificate monitoring commands

Command	Function
? (HELP)	Lists commands and their available options.
CRL	Enters the CRL monitoring menu.
LIST	Lists the certificates on the device.
SCEP	Enters the SCEP monitoring menu.
EXIT	Exits the certificate monitoring menu.

3.2.3.1 crl

Enters the CRL monitoring menu. The commands available in this menu are as follows:

Command	Function
? (HELP)	Lists the commands or their available options.
LIST	Lists the CRLs on the device.
EXIT	Exits the certificate monitoring menu.

3.2.3.1.1 list

Command	Function
EXISTENT	Lists the CRLs written into the device's non-volatile memory.
LOADED	Lists active CRLs.

“LIST EXISTENT”

Displays those CRL lists written into the device's non-volatile memory.

Example:

```
CRL monit+list existent
 A:                   BINTEC.CRL       1273   05/19/11   17:10   Flash
 A:                  WIN2008.CRL       1034   05/26/11   15:28   Flash
```

“LIST LOADED”

Displays active CRL lists.

Example:

```
CRL monit+list loaded
Name
----
WIN2008.CRL
```

3.2.3.2 list

Command	Function
LOADED-CERTIFICATES	Lists certificates that are currently active on the device.
DISK-CERTIFICATES	Lists active certificates saved to disk.
CONFIG-CERTIFICATES	Lists active certificates saved to the configuration.

3.2.3.2.1 list loaded-certificates

Displays active certificates and certificate status.

Example:

```
CERTIFICATES monit+list loaded-certificates
WIN2008.CER (from config)
 Issuer: A:WIN2008.CER
 Status:
               -cn=jorge,dc=pruebas,dc=com
               -Without Private Key
               -CA certificate: A:WIN2008.CER
               -Status:Signature ok.

 Fingerprint md5: E4 EE 9E F1 83 6D AF 2F F8 5A E2 94 52 74 81 6D


 Fingerprint sha1: 14 35 64 A6 42 09 DD 26 AB 67 BF 9A 3F 16 74 D4 7F B9 CB D9

-----------------------------------------------
BINTEC08.CER (from config)
 Issuer: A:WIN2008.CER
 Status:
               -cn=routerjose,dc=com,dc=pruebas
               -Without Private Key
               -CA certificate: A:WIN2008.CER
               -Status:Signature ok.
  Fingerprint md5: E4 EE 9E F1 83 6D AF 2F F8 5A E2 94 52 74 81 6D


 Fingerprint sha1: 14 35 64 A6 42 09 DD 26 AB 67 BF 9A 3F 16 74 D4 7F B9 CB D9
-----------------------------------------------
WIN200EN.CER (from config)
 Issuer: A:WIN2008.CER
 Status:              -cn=integrate,ou=international,o=bintec,l=madrid,s=madrid,c=ES
               -Without Private Key
               -CA certificate: A:WIN2008.CER
               -Status:Signature ok.
  Fingerprint md5: E4 EE 9E F1 83 6D AF 2F F8 5A E2 94 52 74 81 6D


 Fingerprint sha1: 14 35 64 A6 42 09 DD 26 AB 67 BF 9A 3F 16 74 D4 7F B9 CB D9
-----------------------------------------------
WIN200OF.CER (from config)
 Issuer: A:WIN2008.CER
Status:
               -cn=integrate,ou=international,o=bintec,l=madrid,s=madrid,c=ES
               -Without Private Key
               -CA certificate: A:WIN2008.CER
               -Status:Signature ok.
  Fingerprint md5: E4 EE 9E F1 83 6D AF 2F F8 5A E2 94 52 74 81 6D


 Fingerprint sha1: 14 35 64 A6 42 09 DD 26 AB 67 BF 9A 3F 16 74 D4 7F B9 CB D9
-----------------------------------------------
```

3.2.3.2.2 list disk-certificates

Displays certificates saved to disk.

Example:

```
CERTIFICATES monit+list disk-certificates
 A:                  CACHAIN.CER       4076   05/11/11   16:33   Flash
 A:                   TMXENR.CER       1492   05/17/04   13:44   Flash
 A:                   SECTEST.CER      1248   05/17/04   13:47   Flash
 A:                  CENTRAL2.CER      1706   05/17/04   13:48   Flash
```

```
A:                      OF8.CER       1650   05/17/04   13:48   Flash
A:                   STROUTE.CER      1200   05/17/04   13:48   Flash
```

3.2.3.2.3 list config-certificates

Displays certificates saved to the configuration.

Example:

```
CERTIFICATES monit+list config-certificates
ROUTER.CER
CACHAIOF.CER
CACHAIEN.CER
CACHAIN.CER
WIN200OF.CER
WIN200EN.CER
WIN2008.CER
BINTEC08.CER
```

3.2.3.3 scep

Accesses the SCEP monitoring menu. The commands available in this menu are as follows:

Command	Function
? (HELP)	Lists the commands or their available options.
CA-CHAIN-INSTALL	Installs the chain of certificates up to the root CA.
CAPABILITIES	Displays the commands supported by the server.
ENROLL	Executes the enroll protocol for a SCEP group.
INSTALL-CA	Executes the install protocol for a SCEP group.
LIST	Lists SCEP group status.
NEXT-CA-INSTALL	Installs the renewed CA certificates.
EXIT	Exits the certificate monitoring menu.

3.2.3.3.1 ca-chain-install, capabilities, enroll, install-ca, next-ca-install

These commands behave in the same way as the commands described in the section entitled *Obtaining certificates through SCEP* on page 137.

Command history:

Release	Modification
11.01.03	The "*next-ca-install*" command is obsolete as of version 11.01.03.

3.2.3.3.2 list

Displays SCEP group status.

Example:

```
SCEP monit+list
Group: 1, enrolltime: 40 mins (expires in 14 mins),
       retry period: 2 mins (remaining time for next retry: 1 mins), status: IDLE
URL: 192.168.213.119, CGI_PATH: /certsrv/mscep/mscep.dll
CA: win2008
Key Encipherment Cert (Encryption): A:WIN200EN.CER
Digital Signature Cert (Enrollment): A:WIN200OF.CER
There is a valid certificate: A:BINTEC_C.CER
```

Command history:

Release	Modification
10.08.43	As of version 10.08.43, this command shows the enrolltime in seconds and no longer shows the timeleft, which is always 1 minute.
10.09.25	As of version 10.09.25, this command shows the enrolltime in seconds and no longer shows the timeleft, which is always 1 minute.
10.09.24.20.07	As of version 10.09.24.20.07, this command shows, in addition to the enrolltime, the time

Release	Modification
	remaining before the user certificate expires, the retry-period and the time remaining before the next retry (all times are shown in minutes).
11.00.00.02.08	As of version 11.00.00.02.08, this command shows the enrolltime in seconds and no longer shows the timeleft, which is always 1 minute.
11.00.04	As of version 11.00.04, this command shows, in addition to the enrolltime, the time remaining before the user certificate expires, the retry-period and the time remaining before the next retry (all times are shown in minutes).
11.01.00	As of version 11.01.00, this command shows, in addition to the enrolltime, the time remaining before the user certificate expires, the retry-period and the time remaining before the next retry (all times are shown in minutes).

3.2.4 IPsecFT monitoring commands

Access the IPsecFT monitoring submenu from the IPsec monitoring menu by entering *FAULT-TOLERANT* .

```
*p 3
Console Operator
+protocol ip
-- IP protocol monitor --
IP+ipsec
-- IPSec protocol monitor --
IPSec+fault-tolerant
-- Fault tolerant IPSec recovery monitor --
Router1 IPSecFT monitor+
```

The following commands are found in this submenu:

Command	Function
? (HELP)	Lists the available commands or their options.
LIST	Lists information on the protocol.
CLEAR	Resets the protocol counters.
EXIT	Exits the IPsecFT monitoring menu.

3.2.4.1 list

Lists IPsecFT information. The options for this command are:

Command	Function
? (HELP)	Lists the available commands or their options.
ALL	Lists all available information on IPsecFT.
BACKUP-TASK	Lists information about the IPsecFT backup tasks.
LOCAL-TUNNELS	Lists the IPsec sessions that IPsecFT manages locally.
MAIN-TASK	Lists information on the IPsecFT main task.
QUEUE	Lists information on the IPsecFT local queue.
REMOTE-TUNNELS	Lists the remotely managed IPsec sessions.

3.2.4.1.1 list all

Lists all the available information on the protocol.

Example:

```
IPSecFT monitor+list all
Backup task state:
        Running:                  TRUE
        Config change pending:    FALSE
        Current connections:      1
        Accumulated connections:  2
        Unregistered connections: 0
                Connection 1:
                 Time since creation:     0d05h02m13s
                 Current state:           STANDBY
                 Local [address/port]:    192.168.212.219:52912
```

```
                Remote [address/port]:   192.168.212.218:1025
                Internal ID:             0x00000002
                Session ID:              0x1636
                Inactivity timeout:      500 milliseconds
                Inherit condition:       VRRP
                Last packet received:    180555
                Number of tunnels:       0
                Monitored IP 01:         192.168.3.225

List of remote tunnels:
Session ID:              0x1636
        Initialized:          TRUE
        Number of tunnels:    0

Main task state:
        Running:                              TRUE
        Suspended:                            FALSE
        Config change pending:                FALSE
        Accumulated conn retries:             7
        Accumulated ACK with error:           0
          Invalid length:                       0
          Invalid version:                      0
          Invalid type:                         0
          Invalid num seq:                      0
          'not add' flag:                       0
          'not del' flag:                       0
        Accumulated timeouts receiving data: 2
        Accumulated communication errors:    5
        Time since last conn retry:          0d05h02m15s
        Current state:                       STANDBY
        Session ID:                          0x0d6e
        Local [address/port]:                192.168.212.219:1030
        Remote [address/port]:               192.168.212.218:52912
        Inactivity timeout:                  500 milliseconds
        Keepalive period:                    100 milliseconds
        Last packet sent:                    180519
        Number of tunnels:                   1
        Monitored IP 01:                     192.168.3.225

List of local tunnels:
ID 0x84b83a40: ep_src 192.168.3.225(500), ep_dest 192.168.3.100(500), spi 0x84b83a40
               action permit, src 192.168.212.0/23, dst 10.10.2.0/24
        Initialized:          TRUE
        Number of tunnels:    1

Message queue:
        Initialized:             TRUE
        Queue size:              1000
        Available elements:      1000
        Used elements:           0
        Sent to queue:           243880
        Errors sending to queue: 0
        Queue full counter:      0
        Retrieved from queue:    243880
```

The above example shows five large blocks of information which we will now explain:

- Block 1: “**Backup task state**” Reports on IPsecFT backup tasks.

  - “**Running**”: Shows whether the backup is running.

  - “**Config change pending**": Shows whether the backup job has any pending configuration changes.

  - “**Current connections**”: Number of active backup tasks.

  - “**Accumulated connections**”: Number of backup sessions accumulated since the last reset.

- “**Unregistered connections**”: Backup tasks that have not been able to register.

- “**Connection X**”: The following information concerns backup task number X.

- “**Time since creation**": Time since the backup task was created.

- “**Current state**”: Shows the current status of the backup task.

- “**Local [address/port]**”: Source address/port the backup task connected to its corresponding main task.

- “**Remote [address/port]**”: Destination address/port the backup task connected to with its corresponding main task.

- “**Internal ID**”: Internal ID of the backup task.

- “**Session ID**”: Backup task ID.

- “**Inactivity timeout**”: Maximum idle time permitted.

- “**Inherit condition**”: Condition for inheriting the IPsec sessions.

- “**Last packet received**”: Last IPsecFT packet received.

- “**Number of tunnels**”: Number of IPsec sessions this backup task controls.

- “**Monitored IP XX**”: IP address number that this backup task is monitoring.

- Block 2: “**List of remote tunnels**”: Lists IPsec sessions that are controlled by backup tasks.

  - “**Session ID**”: The following information refers to the specified backup task.

  - “**ID 0xXXXXXXXX**”: Displays information on the specified IPsec session.

  - “**Initialized**”: Indicates whether this list of IPsec sessions has initialized.

  - “**Number of tunnels**”: Number of IPsec sessions managed by the specified session.

- Block 3: “**Main task state**”: Lists information on the main IPsecFT task.

  - “**Running**”: Indicates whether the main task is running.

  - “**Suspended**”: Indicates whether the main task is suspended.

  - “**Config change pending**”: Indicates whether the main task has any configuration changes pending.

  - “**Accumulated conn retries**”: Main task reconnection attempts to its corresponding backup task.

  - “**Accumulated ACK with error**”: ACK received with error.

  “**Invalid length**”: ACK received with error due to packet length.

  “**Invalid version**”: ACK received with error due to the protocol version.

  “**Invalid type**”: ACK received with error due to the type of packet received.

  “**Invalid num seq**”: ACK received with error due to sequence number.

  “**'not add' flag**”: ACK received with error due to a notification stating that the specified IPsec session could not be added.

  “**'not del' flag**”: ACK received with error due to a notification stating that the specified IPsec session could not be deleted.

  - “**Accumulated timeouts receiving data**”: Accumulated timeouts while waiting to receive data from the backup task.

  - “**Accumulated communication errors**”: Accumulated communication errors that have occurred.

  - “**Time since last conn retry**”: Time passed since the last reconnection attempt with the backup task.

- “**Current state**”: Current status of the main task.

- “**Session ID**”: Main task ID.

- “**Local [address/port]**”: Source address/port the backup task used in the connection.

- “**Remote [address/port]**”: Destination address/port the backup task connected to.

- “**Inactivity timeout**”: Maximum time permitted without receiving data from the backup task.

- “**Keepalive period**”: Time waited for the next main task action.

- “**Last packet sent**”: Number of packets sent.

- “**Number of tunnels**”: Number of IPsec sessions controlled by the main task.

- “**Monitored IP XX**”: IP address number that this main task is monitoring.

- Block 4: “**List of local tunnels**”: Lists the IPsec sessions controlled by the main task.

  “**ID 0xXXXXXXXX**”: Displays information on the specified IPsec session.

  “**Initialized**”: Indicates whether this list of IPsec sessions has initialized.

  “**Number of tunnels**”: Number of IPsec sessions controlled by the main task.

- Block 5 “**Message queue**”: Lists IPsecFT message queue information.

  “**Initialized**”: Indicates whether or not the message queue is initialized.

  “**Queue size**”: Indicates the size of the message queue.

  “**Available elements**”: Number of available elements in the message queue.

  “**Used elements**”: Number of used elements in the message queue.

  “**Sent to queue**”: Number of elements sent to the message queue.

  “**Errors sending to queue**”: Number of errors sending elements to the queue.

  “**Queue full counter**”: Number of times that the queue is full when trying to enter an element.

  “**Retrieved from queue**”: Number of elements retrieved from the queue.

3.2.4.1.2 LIST BACKUP-TASK

Displays information on IPsecFT backup tasks.

Example:

```
IPSecFT monitor+list backup-task
Backup task state:
        Running:                  TRUE
        Config change pending:    FALSE
        Current connections:      1
        Accumulated connections:  2
        Unregistered connections: 0
                Connection 1:
                 Time since creation:     0d05h42m46s
                 Current state:           STANDBY
                 Local [address/port]:    192.168.212.219:52912
                 Remote [address/port]:   192.168.212.218:1025
                 Internal ID:             0x00000002
                 Session ID:              0x1636
                 Inactivity timeout:      500 milliseconds
                 Inherit condition:       VRRP
                 Last packet received:    204781
                 Number of tunnels:       0
                 Monitored IP 01:         192.168.3.225
```

For more information about the meaning of each field, please refer to the *LIST ALL* monitoring command example in this section.

3.2.4.1.3 list local-tunnels [Filter]

Displays information on those IPsec sessions controlled by the main task and matching the specified filter.

Example:

```
IPSecFT monitor+list local-tunnels
List of local tunnels:
ID 0x84b83a40: ep_src 192.168.3.225(500), ep_dest 192.168.3.100(500), spi 0x84b83a40
               action permit, src 192.168.212.0/23, dst 10.10.2.0/24
        Initialized:          TRUE
        Number of tunnels:    1
```

No filter has been specified in this example, so all IPsec sessions controlled by the main task are shown.

Example:

```
IPSecFT monitor+list local-tunnels 10.10.1.7
List of local tunnels:
ID 0x04483a10: ep_src 192.168.219.225(0), ep_dest 10.10.1.7(0), spi 0x04483a10
               action permit, src 192.168.212.0/23, dst 10.10.1.7/32
ID 0x342c3c50: ep_src 192.168.219.225(0), ep_dest 10.10.1.70(0), spi 0x342c3c50
               action permit, src 192.168.212.0/23, dst 10.10.1.70/32
ID 0x14483a01: ep_src 192.168.219.225(0), ep_dest 10.10.1.71(0), spi 0x14483a01
               action permit, src 192.168.212.0/23, dst 10.10.1.71/32
ID 0x74283ea0: ep_src 192.168.219.225(0), ep_dest 10.10.1.72(0), spi 0x74283ea0
               action permit, src 192.168.212.0/23, dst 10.10.1.72/32
ID 0x84ba3ab4: ep_src 192.168.219.225(0), ep_dest 10.10.1.73(0), spi 0x84ba3ab4
               action permit, src 192.168.212.0/23, dst 10.10.1.73/32
        Initialized:          TRUE
        Number of tunnels:    650
```

In this example, we have specified a filter (10.10.1.7). Now you will be able to see all IPsec sessions controlled by the main task and matching the specified filter - in this case 5 out of 650 sessions.

For further information about the meaning of each field, please refer to the *LIST ALL* monitoring command example in this section.

3.2.4.1.4 list main-task

Displays information on the IPsecFT main task.

Example:

```
IPSecFT monitor+list main-task
Main task state:
        Running:                            TRUE
        Suspended:                          FALSE
        Config change pending:              FALSE
        Accumulated conn retries:           7
        Accumulated ACK with error:         0
          Invalid length:                     0
          Invalid version:                    0
          Invalid type:                       0
          Invalid num seq:                    0
          'not add' flag:                     0
          'not del' flag:                     0
        Accumulated timeouts receiving data: 2
        Accumulated communication errors:   5
        Time since last conn retry:         0d05h46m28s
        Current state:                      STANDBY
        Session ID:                         0x0d6e
        Local [address/port]:               192.168.212.219:1030
        Remote [address/port]:              192.168.212.218:52912
        Inactivity timeout:                 500 milliseconds
        Keepalive period:                   100 milliseconds
        Last packet sent:                   206932
```

```
        Number of tunnels:                  1
        Monitored IP 01:                    192.168.3.225
```

For further information about the meaning of each field, please refer to the *LIST ALL* monitoring command example in this section.

3.2.4.1.5 list queue

Displays information on the IPsecFT message queue.

Example:

```
IPSecFT monitor+list queue
Message queue:
        Initialized:             TRUE
        Queue size:              1000
        Available elements:      1000
        Used elements:           0
        Sent to queue:           273224
        Errors sending to queue: 0
        Queue full counter:      0
        Retrieved from queue:    273224
```

For further information about the meaning of each field, please refer to the *LIST ALL* monitoring command example in this section.

3.2.4.1.6 list remote-tunnels [Filter]

Displays information on those IPsec sessions controlled by backup tasks and matching the specified filter.

Example:

```
IPSecFT monitor+list remote-tunnels
List of remote tunnels:
Session ID:              0x0d6e
        ID 0x3caf53e8: ep_src 192.168.3.225(500), ep_dest 192.168.3.100(500), spi 0x3caf53e8
                       action permit, src 192.168.212.0/23, dst 10.10.2.0/24
        Initialized:          TRUE
        Number of tunnels:    1
```

We have not specified a filter in this example so all IPsec sessions controlled by the backup tasks are shown.

Example:

```
IPSecFT monitor+list remote-tunnels 10.10.1.7
List of remote tunnels:
Session ID:              0xca8f
        ID 0x04483a10: ep_src 192.168.219.225(0), ep_dest 10.10.1.7(0), spi 0x04483a10
                       action permit, src 192.168.212.0/23, dst 10.10.1.7/32
        ID 0x342c3c50: ep_src 192.168.219.225(0), ep_dest 10.10.1.70(0), spi 0x342c3c50
                       action permit, src 192.168.212.0/23, dst 10.10.1.70/32
        ID 0x14483a01: ep_src 192.168.219.225(0), ep_dest 10.10.1.71(0), spi 0x14483a01
                       action permit, src 192.168.212.0/23, dst 10.10.1.71/32
        ID 0x74283ea0: ep_src 192.168.219.225(0), ep_dest 10.10.1.72(0), spi 0x74283ea0
                       action permit, src 192.168.212.0/23, dst 10.10.1.72/32
        ID 0x84ba3ab4: ep_src 192.168.219.225(0), ep_dest 10.10.1.73(0), spi 0x84ba3ab4
                       action permit, src 192.168.212.0/23, dst 10.10.1.73/32
        Initialized:          TRUE
        Number of tunnels:    650
```

In this example, we have specified a filter (10.10.1.7). Now you will be able to see all IPsec sessions controlled by the backup tasks and matching the specified filter - in this case 5 out of 650 sessions in a single backup task.

For further information about the meaning of each field, please refer to the *LIST ALL* monitoring command example in this section.

3.2.4.2 clear

Resets IPsecFT monitoring counters. The options for this command are as follows:

Command	Function

? (HELP)	Lists the commands or their available options.
ALL	Resets any counters that allow resetting.
BACKUP-TASK	Resets any backup task counters that allow resetting.
MAIN-TASK	Resets any main task counters that allow resetting.
QUEUE	Resets IPsecFT message queue counters that allow resetting.

3.2.4.2.1 clear all

Resets any of the IPsecFT monitoring counters that allow resetting.

Example:

```
IPSecFT monitor+clear all
```

3.2.4.2.2 clear backup-task

Resets any of the IPsecFT monitoring backup task counters that allow resetting.

Example:

```
IPSecFT monitor+clear backup-task
```

3.2.4.2.3 clear main-task

Resets any of the IPsecFT monitoring main task counters that allow resetting.

Example:

```
IPSecFT monitor+clear main-task
```

3.2.4.2.4 clear queue

Resets any of the IPsecFT monitoring message queue counters that allow resetting.

Example:

```
IPSecFT monitor+clear queue
```

3.2.5 GDOI group monitoring commands

Access the Group Domain Of Interpretation (GDOI) monitoring submenu from the IPsec monitoring menu by entering *GDOI GROUP [ID]*. This command is only valid when the device is configured as a GDOI key server.

```
*p 3
Console Operator


+pro ip

-- IP protocol monitor --

IP+ipsec

-- IPSec protocol monitor --

IPSec+gdoi group 123

-- GDOI group monitor --
IPSec GDOI monitor+
```

The following commands are found in this submenu:

Command	Function
? (HELP)	Lists the available commands or their options.
LIST	Lists GDOI parameters.
REKEY	Forces a rekey.
EXIT	Exits the GDOI group monitoring menu.

3.2.5.1 List

Lists information about the GDOI group:

Example:

```
IPSec GDOI monitor+list group

KEK lifetime: 0h5m0s,next KEK in 0h4m19s
Retransmit 0 out of 3, time-out 10, next ret 0
Cooki: 0x6d8894d8a6d180f2
Cookr: 0x3f2e2684c963a6c

                    GDOI Server registered clients:
------------------------------------------------------------------------
   Address 20.0.2.1:848
     Last Register was 0h5m38s ago
     Rekeys without ACK 0

                              GDOI Server SAs:
------------------------------------------------------------------------
    spi 3574e130 lifetime 2h0m0s dies in 1h52m43s rekeys in 1h50m43s

IPSec GDOI monitor+
```

3.2.5.2 Rekey

Sends a REKEY packet to GDOI group members:

Example:

```
IPSec GDOI monitor+rekey
IPSec GDOI monitor+
```

3.2.6 Problem diagnosis in IKE

This section will present examples of typical problems that often occur during IKE negotiation due to configuration errors. Identifying which phase the negotiation is currently in is very important. To find out, all you need to do is look at the number associated with the message header that caused the error. If this number is 0, then the message is from phase I. If it is a number other than zero, it is from phase II. The message that produced the error is usually the one preceding the notification message indicating the error. For example:

```
172.24.51.57: (HDR 0)(HDR sa)(prop 1 isakmp #1)(trans 1 id=1)(encrypt des)
  (hash sha)(grp desc 1)(auth rsa)(life sec)(duration 600)(vendor 14)
172.24.78.15:
  (* --------- Creating ISAKMP NEG ---------)(# 57(0x39))(HDR 24343432)
  (notif isakmp no proposal chosen)
```

The message causing the error was sent by device 172.24.51.57, whose HDR has the "0" identifier. This means that the error occurred in the first phase of the negotiation.

Another very important thing to know is who commenced negotiations (i.e., who was the initiator).

3.2.6.1 The device does not initiate the negotiation

Origin

The access control list (ACL) is configured incorrectly.

This message occurs because the device hasn't been able to match the packet that should trigger the negotiation with an IPsec ACL entry.

Solution

Check the ACL parameters.

Addresses: Source and destination. (Be careful with subnets).

Mask.

Protocol.

Ports. Source and destination.

Template: The corresponding dynamic template must be mapped.

If you still cannot find the source of the error, take a look at the *LIST ACCESS-LIST ENTRIES* monitoring command output to see if the hits in the corresponding entry are increasing.

3.2.6.2 notif isakmp no proposal chosen. Phase I

Initiator: 172.24.51.57

```
172.24.51.57: (HDR 0)(HDR sa)(prop 1 isakmp #1)(trans 1 id=1)(encrypt des)
  (hash sha)(grp desc 1)(auth rsa)(life sec)(duration 600)(vendor 14)
172.24.78.15:
  (* --------- Creating ISAKMP NEG ---------)(# 57(0x39))(HDR 0)
  (notif isakmp no proposal chosen)
```

Origin

The ISAKMP template is not configured correctly.

This message occurs because device 172.24.78.15 was unable to accept any of the proposals from device 172.24.51.57. In this negotiation phase, the proposals received are compared with those configured in the ISAKMP template.

Solution

Check the ISAKMP template parameters.

Authentication method: RSA_SIGNATURE, PRESHARED, etc.

Encryption system: DES, TDES, etc.

Authentication system: SHA1, MD5, etc.

Type of lifetime: Seconds, Kbytes, both.

Group: 1 or 2.

3.2.6.3 notif isakmp payload malformed. Phase I

Initiator: 172.24.51.57

```
172.24.51.57: (HDR 0)(HDR sa)(prop 1 isakmp #1)(trans 1 id=1)(encrypt des)
  (hash md5)(grp desc 1)(auth presh)(life sec)(duration 600)(vendor 14)
172.24.78.15:
  (* --------- Creating ISAKMP NEG ---------)(# 67(0x43))
  (* --------- Matching template ---------)(# 20(0x14))(HDR 0)(HDR sa)
  (prop 1 isakmp #1)(trans 1 id=1)(encrypt des)(hash md5)(grp desc 1)(auth presh)
  (life sec)(duration 600)
172.24.51.57: (HDR 0)(HDR keyx)(HDR nonce)
172.24.78.15: (HDR 0)(HDR keyx)(HDR nonce)
  (* --------- Creating ISAKMP SA ---------)
172.24.51.57: (HDR 0)(id none prot=148 port=9841)(# 0x3c068321)(HDR 75 0)
172.24.78.15: (HDR 0)(notif isakmp payload malformed)
```

Origin

The pre-shared key is not configured correctly.

This message occurs because device 172.24.78.15 was unable able to decode the encrypted message sent by device 172.24.51.57. In fact, if we take a closer look at the erroneous message, we can see that it contains several strange parameters: unknown identifier (where the protocol and port are different to the ones that were configured) followed by an unknown header, .hdr 75 0.

Solution

Check the pre-shared key and the ip_address – key, hostname-key associations.

3.2.6.4 notif esp no proposal chosen. Phase II

Initiator: 172.24.51.57

```
172.24.51.57: (HDR 53da7bd5)(HDR hash)(HDR sa)(prop 1 esp #2)
  (# -786612676(0xd11d3e3c))(trans 1 id=des)(life sec)(duration 300)
  (life kbytes)(duration 100000)(encap tunnel)(auth alg md5)(trans 2 id=des)
  (life sec)(duration 300)(life kbytes)(duration 100000)(encap tunnel)
  (auth alg sha)(prop 2 ah #2)(# -786612676(0xd11d3e3c))(trans 1 id=md5)
  (life sec)(duration 300)(life kbytes)(duration 100000)(encap tunnel)
  (auth alg md5)(trans 2 id=sha)(life sec)(duration 300)(life kbytes)
  (duration 100000)(encap tunnel)(auth alg sha)(HDR nonce)
  (id addr4 prot=0 port=0)(# 0xac183339)(id addr4 prot=0 port=0)(# 0xac184e0f)
172.24.78.15:
  (* --------- Creating ISAKMP SA id ---------)(# -583852704(0xdd331d60))
  (HDR dd331d60)(HDR hash)(notif esp no proposal chosen)
```

Origin

The ISAKMP template is not configured correctly.

This message occurs because device 172.24.78.15 was unable to accept any of the proposals from device 172.24.51.57. In this negotiation phase, the proposals received are compared with the ones configured in the dynamic template associated with the corresponding ACL.

Solution

Check the dynamic template parameters.

Type of encapsulation: Tunnel or transport.

Encryption system: DES, TDES, etc.

Authentication system: SHA1, MD5, etc.

Type of lifetime: Seconds, Kbytes, both.

PFS: Check that the remote device supports PFS.

3.2.6.5 notif esp invalid id inform. Phase II

Initiator: 172.24.51.57

```
172.24.78.15: (HDR 0)(id addr4 prot=17 port=500)(# 0xac184e0f)(HDR hash)
  (* --------- Creating ISAKMP SA id ---------)(# 785093687(0x2ecb9437))
172.24.51.57: (HDR 2ecb9437)(HDR hash)(HDR sa)(prop 1 esp #2)
  (# 291357516(0x115dc34c))(trans 1 id=des)(life sec)(duration 300)(life kbytes)
  (duration 100000)(encap tunnel)(auth alg md5)(trans 2 id=des)(life sec)
  (duration 300)(life kbytes)(duration 100000)(encap tunnel)(auth alg sha)
  (prop 2 ah #2)(# 291357516(0x115dc34c))(trans 1 id=md5)(life sec)
  (duration 300)(life kbytes)(duration 100000)(encap tunnel)(auth alg md5)
  (trans 2 id=sha)(life sec)(duration 300)(life kbytes)(duration 100000)
  (encap tunnel)(auth alg sha)(HDR nonce)(id addr4 prot=0 port=0)(# 0xac183339)
  (id addr4 prot=16 port=0)(# 0xac184e0f)
172.24.78.15:
  (* --------- Creating ISAKMP SA id ---------)(# 1537079449(0x5b9df899))
  (HDR 5b9df899)(HDR hash)(notif esp invalid id inform)
```

Origin

The ACL is configured incorrectly.

This message occurs because device 172.24.78.15 has not been able to accept the client identifier from device 172.24.51.57 (id addr4 prot=0 port=0)(# 0xac183339) (id addr4 prot=16 port=0)(# 0xac184e0f). In this negotiation phase, the proposals of the identifiers received are compared with the ones configured in the ACL.

Solution

Check the ACL parameters.

Addresses: Source and destination. (Be careful with subnets).

Mask.

Protocol.

Ports. Source and destination.

Template: The corresponding dynamic template must be mapped.

3.2.6.6 notif isakmp invalid cert authority. Phase I. Initiator A

Initiator: 172.24.78.15

```
172.24.78.15: (HDR 0)(HDR keyx)(HDR nonce)
172.24.51.57: (HDR 0)(HDR keyx)(HDR nonce)(certreq x509sig CERTREG 8)
172.24.78.15:
  (* --------- Creating ISAKMP SA ---------)(HDR 0)
  (notif isakmp invalid cert authority)
```

Origin

The ISAKMP template is not configured correctly.

This message occurs because device 172.24.78.15 was unable to find the CA in the corresponding ISAKMP template.

Solution

Check the ISAKMP template parameters.

Name of the CA.

Check that the CA name corresponds to a file in the device:

```
Router CERTIFICATES config>list disk-certificates
```

Check that the CA name corresponds to one of the configured CA names:

```
Router CERTIFICATES config>list config-certificates
```

3.2.6.7 notif isakmp invalid cert authority. Phase I. Initiator B

Initiator: 172.24.51.57

```
172.24.78.15: (HDR 0)(HDR keyx)(HDR nonce)(certreq x509sig CERTREG 6)
  (* --------- Creating ISAKMP SA ---------)
172.24.51.57: (HDR 0)(id der_dn port=0 CERTREG 7)(cert x509sig CERTREG 8)
  (HDR sig)(certreq x509sig CERTREG 9)
172.24.78.15: (HDR 0)(notif isakmp invalid cert authority)
```

Origin

The ISAKMP template is not configured correctly.

This message occurs because device 172.24.78.15 was unable to find the CA configured on the certificate it received (CERTREG 9, in the above example), among the CAs configured on the ISAKMP templates.

Solution

Check the ISAKMP template parameters and compare them to the monitoring command output.

```
Router IPSec+list certificate_number 9
```

Name of the CA.

Check that the CA name corresponds to a device file:

```
Router CERTIFICATES config>list disk-certificates
```

Check that the CA name matches one of the configured CAs:

```
Router CERTIFICATES config>list config-certificates
```

3.2.6.8 notif isakmp invalid cert. Phase I

Initiator: 172.24.51.57

```
172.24.51.57: (HDR 0)(HDR keyx)(HDR nonce)
172.24.78.15: (HDR 0)(HDR keyx)(HDR nonce)(certreq x509sig CERTREG 14)
  (* --------- Creating ISAKMP SA ---------)
172.24.51.57: (HDR 0)(id der_dn port=0 CERTREG 15)(cert x509sig CERTREG 16)
  (HDR sig)(certreq x509sig CERTREG 17)
172.24.78.15: (HDR 0)(notif isakmp invalid cert)
```

Origin

The certificate that was received is invalid.

Solution

Use the monitoring command to check that the received certificate is correct:

```
Router IPSec+list certificate_number 16
```

Check the following parameters:

Validity period.

The issuer in the certificate matches the required CA.

```
Router IPSec+list certificate_number 14
```

The certificate may have been signed incorrectly.

3.2.6.9 notif isakmp cert unavailable. Phase I

Initiator: 172.24.51.57

```
172.24.51.57: (HDR 0)(HDR keyx)(HDR nonce)
172.24.78.15: (HDR 0)(HDR keyx)(HDR nonce)(certreq x509sig CERTREG 0)
  (* --------- Creating ISAKMP SA ---------)
172.24.51.57: (HDR 0)(id der_dn port=0 CERTREG 1)(cert x509sig CERTREG 2)
  (HDR sig)(certreq x509sig CERTREG 3)
172.24.78.15: (HDR 0)(notif isakmp cert unavailable)
```

Origin

There is no user certificate loaded for device 172.24.78.15 to send to the 172.24.51.57 end.

Solution

Check that a certificate has been loaded for the required CA.

First of all, check which CA is required.

```
Router IPSec+list certificate_number 3
```

If the required CA matches the one that was sent, run a list of the ISAKMP templates and check the result. This should tell us what the problem is.

If the required CA does not match the one that was sent, consult the CERTIFICATES menu to see whether a certificate from this CA has been loaded.

```
Router CERTIFICATES config>list loaded-certificates
```