IP Tunnels

In This Section

This section provides an overview of IP Security (IPSec) software features for the IPSec ISA.

Topics in this section include:

- IP Tunnels Overview on page 416
  - → Tunnel ISAs on page 419
  - → Operational Conditions on page 426
  - → QoS Interactions on page 427
  - → OAM Interactions on page 427
  - → Redundancy on page 427
  - → Statistics Collection on page 429
  - → Security on page 429
  - → IKEv2 on page 432
  - → SHA2 Support on page 434
  - → Using Certificates For IPSec Tunnel Authentication on page 443
  - → Trust-Anchor-Profile on page 444
  - → Certificate Management Protocol Version 2 (CMPv2) on page 448
  - → OCSP on page 449
  - → Video Wholesale Example on page 450
  - → Multi-Chassis IPSec Redundancy Overview on page 451
  - → IPSec Deployment Requirements on page 460
  - → IKEv2 Remote-Access Tunnel on page 462

IP Tunnels Overview

This section discusses IP Security (IPSec), GRE tunneling, and IP-IP tunneling features supported by the MS-ISA. In these applications, the MS-ISA functions as a resource module for the system, providing encapsulation and (for IPSec) encryption functions. The IPSec encryption functions provided by the MS-ISA are applicable for many applications including: encrypted SDPs, video wholesale, site-to-site encrypted tunnel, and remote access VPN concentration.

Figure 33 shows an example of an IPSec deployment, and the way this would be supported inside a 7750. GRE and IP-IP tunnel deployments are very similar. IP tunnels have two flavors GRE/IP-IP, in all but a few area the information for IP Tunnels applies to both types.

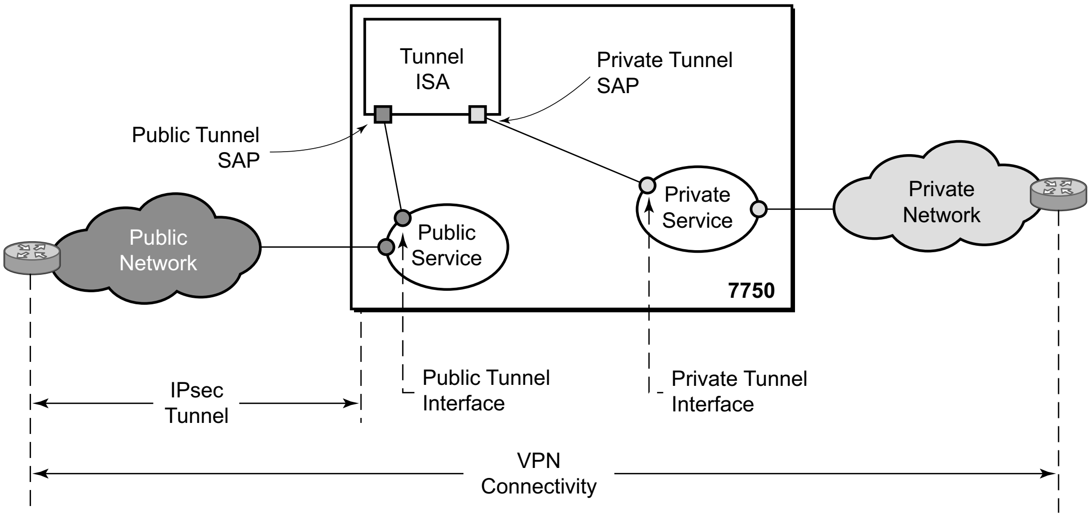

Figure 33: 7750 IPSec Implementation Architecture

Figure 33, the public network is typically an “insecure network” (for example, the public Internet) over which packets belonging to the private network in the diagram cannot be transmitted natively. Inside the 7750, a public service instance (IES or VPRN) connects to the public network and a private service instance (typically a VPRN) connects to the private network.

The public and private services are typically two different services, and the MS-ISA is the only “bridge” between the two. Traffic from the public network may need to be authenticated and encrypted inside an IPSec tunnel to reach the private network. In this way, the authenticity/confidentiality/integrity of accessing the private network can be enforced.If authentication and confidentiality are not important then access to the private network may alternatively be provided through GRE or IP-IP tunnels.

The MS-ISA provides a variety of encryption features required to establish bi-directional IPSec tunnels including:

Control Plane:

- Manual Keying
- Dynamic Keying: IKEv1/v2
- IKEv1 Mode: Main and Aggressive
- Authentication: Pre-Shared-Key /xauth with RADIUS support/X.509v3 Certificate/EAP
- Perfect Forward Secrecy (PFS)
- DPD
- NAT-Traversal
- Security Policy

Data Plane:

- ESP (with authentication) Tunnel mode
- Authentication Algorithm: MD5/SHA1/SHA256/SHA384/SHA512/AES-XCBC
- Encryption Algorithm: DES/3DES/AES128/AES192/AES256
- DH-Group: 1/2/5/14/15
- Anti-Replay Protection
- N:M IPSec ISA card redundancy

Note: SR OS will use a configured authentication algorithm in an ike-policy for Pseudorandom Function (PRF).

There are two types of tunnel interfaces and SAPs:

- Public tunnel interface: configured in the public service; outgoing tunnel packets have a source IP address in this subnet
- Public tunnel SAP: associated with the public tunnel interface; a logical access point to the MS-ISA card in the public service
- Private tunnel interface: configured in the private service; can be used to define the subnet for remote access IPSec clients.
- Private tunnel SAP: associated with the private tunnel interface, a logical access point to the MS-ISA card in the private service

Traffic flows to and through the MS-ISA card as follows:

- In the upstream direction, the encapsulated (and possibly encrypted) traffic is forwarded to a public tunnel interface if its destination address matches the local or gateway address of an IPSec tunnel or the source address of a GRE or IP-IP tunnel. Inside the MS-ISA card, encrypted traffic is decrypted, the tunnel header is removed, the payload IP packet is delivered to the private service, and from there, the traffic is forwarded again based on the destination address of the payload IP packet.

- In the downstream direction, unencapsulated/clear traffic belonging to the private service is forwarded into the tunnel by matching a route with the IPSec/GRE/IP-IP tunnel as next-hop. The route can be configured statically, learned by running OSPF on the private tunnel interface (GRE tunnels only), learned by running BGP over the tunnel (IPSec and GRE tunnels only), or learned dynamically during IKE negotiation (IPSec only). After clear traffic is forwarded to the MS-ISA card, it is encrypted if required, encapsulated per the tunnel type, delivered to the public service, and from there, the traffic is forwarded again based on the destination address of the tunnel header.

Tunnel ISAs

A tunnel-group is a collection of MS-ISAs (each having mda-type **isa-tunnel**) configured to handle the termination of one or more IPSec, GRE and/or IP-IP tunnels. Two example tunnel-group configurations are shown below:

```
config isa
     tunnel-group 1 create
          primary 1/1
          backup 2/1
          no shutdown
          exit


config isa
     tunnel-group 2 create
          multi-active
          mda 3/1
          mda 3/2
          no shutdown
```

A GRE, IP-IP, or IPSec tunnel belongs to only one tunnel group. There are two types of tunnel groups:

- A single-active tunnel-group can have one tunnel-ISA designated as primary and optionally one other tunnel-ISA designated as backup. If the primary ISA fails the affected failed tunnels are re-established on the backup (which is effectively a cold standby) if it is not already in use as a backup for another tunnel-group.
- A multi-active tunnel-group can have multiple tunnel-ISAs designated as primary. This is only supported on 7750 SR7/SR12/SR12E with chassis mode D or 7450 mixed mode with IOM3.

The show isa tunnel-group allows the operator to view information about all configured tunnelgroups.This command displays the following information for each tunnel-group: group ID, primary tunnel-ISAs, backup tunnel-ISAs, active tunnel-ISAs, admin state and oper state.

Public Tunnel SAPs

A VPRN or IES service (the delivery service) must have at least one IP interface associated with a public tunnel SAP to receive and process the following types of packets associated with GRE, IP-IP and IPSec tunnels:

- GRE (IP protocol 47)
- IP-IP (IP protocol 4)
- IPSec ESP (IP protocol 50)
- IKE (UDP)

The public tunnel SAP type has the format tunnel-*tunnel-group*.public:*index*, as shown in the following CLI example.

```
*A:Dut-C>config>service# info
----------------------------------------------
        customer 1 create
            description "Default customer"
        exit
        ies 1 customer 1 create
            interface "public" create
                address 64.251.12.1/24
                tos-marking-state untrusted
                sap tunnel-1.public:200 create
                exit
            exit
            no shutdown
        exit
        vprn 2 customer 1 create
            route-distinguisher 1.1.1.1:65007
            interface "greTunnel" tunnel create
                address 10.0.0.1/24
                dhcp
                    no shutdown
                exit
                sap tunnel-1.private:210 create
                    ip-tunnel "toCel" create
                        dest-ip 10.0.0.2
                        gre-header
                        source 64.251.12.88
                        remote-ip 64.251.12.2
                        backup-remote-ip 64.251.12.22
                        delivery-service 1
                        no shutdown
                    exit
                exit
            exit
            no shutdown
        exit
----------------------------------------------
*A:Dut-C>config>service#
```

Private Tunnel SAPs

The private service must have an IP interface to a GRE, IP-IP, or IPSec tunnel in order to forward IP packets into the tunnel, causing them to be encapsulated (and possibly encrypted) per the tunnel configuration and to receive IP packets from the tunnel after the encapsulation has been removed (and decryption). That IP interface is associated with a private tunnel SAP.

The private tunnel SAP has the format tunnel-*tunnel-group*.private:*index*, as shown in the following CLI example where a GRE tunnel is configured under the SAP.

```
*A:Dut-A# show ip tunnel
===============================================================================
IP Tunnels
===============================================================================
TunnelName                       SapId                          SvcId      Admn
 Local Address                                                  DlvrySvcId Oper
  OperRemoteAddress
-------------------------------------------------------------------------------
tun-1-gre-tunnel                 tunnel-1.private:1             201        Up
 141.1.1.2                                                      1201       Up
  41.1.1.2
-------------------------------------------------------------------------------
IP Tunnels: 1
===============================================================================
```

IP Interface Configuration

In the configuration example of the previous section the IP address 10.0.0.1 is the address of the GRE tunnel endpoint from the perspective of payload IP packets. This address belongs to the address space of the VPRN 1 service and will not be exposed to the public IP network carrying the GRE encapsulated packets. An IP interface associated with a private tunnel SAP does not support unnumbered operation.

It is possible to configure the IP MTU (M) of a private tunnel SAP interface. This sets the maximum payload IP packet size (including IP header) that can be sent into the tunnel – for example, it applies to the packet size before the tunnel encapsulation is added. When a payload IPv4 packet that needs to be forwarded into the tunnel is larger than M bytes the payload packet is IP fragmented (prior to tunnel encapsulation) if the DF bit is clear, otherwise the packet is discarded. When a payload IPv6 packet that needs to be forwarded into the tunnel is larger than M bytes the packet is discarded if its size is less than 1280 bytes otherwise it is forwarded and encapsulated intact.

GRE and IP-IP Tunnel Configuration

To bind an IP/GRE or IP-IP tunnel to a private tunnel SAP, the **ip-tunnel** command should be added under the SAP. To configure the tunnel as an IP/GRE tunnel, the **gre-header** command must be present in the configuration of the **ip-tunnel**. To configure the tunnel as an IP-IP tunnel, the **ip-tunnel** configuration should have the **no gre-header** command. When configuring a GRE or IP-IP tunnel, the **dest-ip** command specifies an IPv4 or IPv6 address (private) of the remote tunnel endpoint. A tunnel can have up to 16 dest-ip addresses. If any of the **dest-ip** addresses are not contained by a subnet of the local private endpoint then the tunnel will not come up. In the CLI sub-tree under **ip-tunnel**, there are commands to configure the following:

- The source address of the GRE or IP-IP tunnel- This is the source IPv4 address of GRE or IP-IP encapsulated packets sent by the delivery service. It must be an address in the subnet of the associated public tunnel SAP interface.
- The remote IP address - If this address is reachable in the delivery service (there is a route) then this is the destination IPv4 address of GRE or IP-IP encapsulated packets sent by the delivery service.
- The backup remote IP address- If the remote IP address of the tunnel is not reachable then this is the destination IPv4 address of GRE or IP-IP encapsulated packets sent by the delivery service.
- The delivery service- This is the id or name of the IES or VPRN service where GRE or IP-IP encapsulated packets are injected and terminated. The delivery service can be the same service where the private tunnel SAP interface resides.
- The DSCP marking in the outer IP header of GRE encapsulated packets- If this is not configured then the default is to copy the DSCP from the inner IP header to the outer IP header.

A private tunnel SAP can have only one ip-tunnel sub-object (one GRE or IP-IP tunnel per SAP).

The show ip tunnel displays information about a specific IP tunnel or all configured IP tunnels. The following information is provided for each tunnel:

- service ID that owns the tunnel
- private tunnel SAP that owns the tunnel
- tunnel name, source address
- remote IP address
- backup remote IP address
- local (private) address
- destination (private) address
- delivery service
- dscp
- admin state
- oper state
- type (GRE or IP-IP)

The following is an example of the output of the **show ip tunnel <tunnel-name>** command.

```
A:config>service>vprn>if>sap>ip-tunnel# show ip tunnel "ipv6-gre"
===============================================================================
IP Tunnel Configuration Detail
===============================================================================
Service Id       : 1                     Sap Id           : tunnel-1.private:1
Tunnel Name      : ipv6-gre
Description      : None
GRE Header       : Yes                   Delivery Service : 2
GRE Keys Set     : False
GRE Send Key     : N/A                   GRE Receive Key  : N/A
Admin State      : Up                    Oper State       : Up
Source Address   : 2002::1:2:3:4
Remote Address   : 3ffe:1::2
Backup Address   : (Not Specified)
Oper Remote Addr : 3ffe:1::2
DSCP             : ef
Reassembly       : inherit
Clear DF Bit     : false                 IP MTU           : max
Encap IP MTU     : 1400
Pkt Too Big      : true
Pkt Too Big Numb*: 100                   Pkt Too Big Intvl: 10 secs
Oper Flags       : None
Last Oper Changed: 02/09/2015 15:22:38
Host MDA         : 1/2

-------------------------------------------------------------------------------
Target Address Table
-------------------------------------------------------------------------------
Destination IP                           IP Resolved Status
-------------------------------------------------------------------------------
172.16.1.2                               Yes
2001:abcd::2                             Yes
-------------------------------------------------------------------------------

===============================================================================
IP Tunnel Statistics: ipv6-gre
===============================================================================
Errors Rx        : 0                     Errors Tx        : 0
Pkts Rx          : 0                     Pkts Tx          : 0
Bytes Rx         : 0                     Bytes Tx         : 0
Key Ignored Rx   : 0                     Too Big Tx       : 0
Seq Ignored Rx   : 0
Vers Unsup. Rx   : 0
Invalid Chksum Rx: 0
Key Mismatch Rx  : 0
===============================================================================

===============================================================================
Fragmentation Statistics
===============================================================================
Encapsulation Overhead                 : 44
Pre-Encapsulation
    Fragmentation Count                : 0
    Last Fragmented Packet Size        : 0
Post-Encapsulation
    Fragmentation Count                : 0
    Last Fragmented Packet Size        : 0
===============================================================================
===============================================================================
```

IP Fragmentation and Reassembly for IP Tunnels

An IPSec, GRE or IP-IP tunnel packet that is larger than the IP MTU of some interface in the public network must either be discarded (if the Do Not Fragment (DF) bit is set in the outer IP header) or fragmented. If the tunnel packet is fragmented, then it is up to the destination tunnel endpoint to reassemble the tunnel packet from its fragments. Starting in Release 10, IP reassembly can be enabled for all the IPSec, GRE, and IP-IP tunnels belonging to a tunnel-group. For IP-IP and GRE tunnels, the reassembly option is also configurable on a per-tunnel basis so that some tunnels in the tunnel-group can have reassembly enabled, and others can have the extra processing disabled. When reassembly is disabled for a tunnel, all received fragments belonging to the tunnel are dropped.

To avoid public network fragmentation of IPSec, GRE, or IP-IP packets belonging to a particular tunnel, one possible strategy is to fragment IPv4 payload packets larger than a specified size M at entry into the tunnel (before encapsulation and encryption if applicable). The size M is configurable using the **ip-mtu** command under the **ip-tunnel** or **ipsec-tunnel**/**tunnel-template** configuration.

If the payload IPv4 packets are all M bytes or less in length then it is guaranteed that all resulting tunnel packets will be less than M+N bytes in length, if N is the maximum overhead added by the tunneling protocol. If M+N is less than the smallest interface IP MTU in the public network, fragmentation will be avoided. In some cases, some of the IPv4 payload packets entering a tunnel may have their DF bit set. And if desired, the SR OS supports the option (also configurable on a per-tunnel basis) to clear the DF bit in these packets so that they can be fragmented.

The system allows users to configure an **encapsulated-ip-mtu** for a given tunnel under an **ip-tunnel** or **ipsec-tunnel**/**tunnel-template** configuration. This represents the maximum size of the encapsulated tunnel packet. After encapsulation, If the IPv4 or IPv6 tunnel packet size exceeds the configured **encapsulated-ip-mtu**, then the system will fragment the packet against the **encapsulated-ip-mtu.**
The following is a description of system behavior about fragmentation:

- Private Side — If the size, before encapsulation, of the IPv4 or IPv6 packet entering the tunnel is larger than the ip-mtu configured under ip-tunnel or ipsec-tunnel/tunnel template:
  - → IPv4 payload packet:
    - – If the DF bit is not set in the packet or if the **clear-df-bit** command is configured, then the system fragments the packet against the ip-mtu configured under ip-tunnel or ipsec-tunnel/tunnel-template.
    - – Otherwise, the system drops the packet and sends back an ICMP error Fragmentation required and DF flag set, with the suggested MTU set as the ip-mtu.

→ IPv6 payload packet:

  – If the packet size >1280 bytes, the system drops the packet and sends back an ICMPv6 Packet Too Big (PTB) message with the suggested MTU set as the ip-mtu.

  – If the packet size<=1280 bytes, the system will forward the packet into the tunnel.

- Public Side — This applies to both ESP and IKE packets, IPv4 and IPv6.

  → If the ESP/IKE packet is larger than the encapsulated-ip-mtu, then the system fragments the packet against the encapsulated-ip-mtu.

Operational Conditions

A tunnel group that is in use cannot be deleted. In single-active mode, changes to the primary ISA are allowed only in when the tunnel group is in a shutdown state. Change to the backup ISA (or the addition of a backup ISA) is allowed at any time unless the ISA is currently active for this tunnel group. When the backup module is active, changing the primary module is allowed without shutting down the tunnel group. If it is part of a multi-chassis configuration, you cannot change the mode until it is removed from this configuration as well.

A shutdown of tunnel-group is required to do the following:

- Change the mode between multi-active and single-active.
- Change the primary-isa in single-active mode.
- Change the active-mda-number in multi-active mode.

In multi-active mode, if the active member ISA goes down, system will replace it with backup ISA; however, if there is no backup ISA, the tunnel-group will be “oper-down”. A multi-active tunnel-group with MC-IPSec enabled cannot be changed into single active-mode unless it is removed from MC-IPSec configuration.

Changes to the ipsec-transform/ike-policy in-use are not allowed.

The public interface address can be changed at any time. However, if changed, tunnels that were configured to use it will require a configuration change. If the public subnet changed is still using an old subnet, the tunnels will be in an operationally down state until their configuration is corrected. The public service cannot be deleted while tunnels are configured to use it. A public service is the IES or VPRN service that hold the regular interface that connects the node to the public network. A private service connects to the private protected service.

A tunnel group ID or tag cannot be changed. To remove an tunnel group instance, it must be in a shutdown state (both front-door and back-door).

A change to the security policy is not allowed while a tunnel is active and using the policy.

The tunnel local-gateway-address, peer address, or delivery router parameters cannot be changed while the tunnel is operationally up (shutdown will make it both admin down and operationally down).

A tunnel security policy cannot be changed while the tunnel is operationally up. An IPSec transform policy or ike-policy assignments to a tunnel requires the tunnel to be shutdown.

QoS Interactions

The MS-ISA can interact with the queuing functions on the IOM through the ingress/egress QoS provisioning in the IES or IP VPN service where the IPSec session is bound. Multiple IPSec sessions can be assigned into a single IES or VPRN service. In this case, QoS defined at the IES or VPRN service level, is applied to the aggregate traffic coming out of or going into the set of sessions assigned to that service.

In order to keep marking relevant in the overall networking design, the ability to translate DSCP bit marking on packets into DSCP bit markings on the IPSec tunneled packets coming out of the tunnel is supported.

OAM Interactions

The MS-ISA is IP-addressed by an operator-controlled IP on the public side. That IP address can be used in Ping and Traceroute commands and the ISA can either respond or forward the packets to the CPM.

For static LAN-to-LAN tunnel, in multi-active mode, a ping requests to public tunnel address would not be answered if the source address is different from the remote address of the static tunnel.

The private side IP address is visible. The status of the interfaces and the tunnels can be viewed using show commands.

Traffic that ingresses or egresses an IES or VPRN service associated with certain IPSec tunnels can be mirrored like other traffic.

Mirroring is allowed per interface (public) or IPSec interface (private) side. A filter mirror is allowed for more specific mirroring.

Redundancy

In single-active mode, every tunnel group can be configured with primary and backup ISAs. An ISA can be used as a backup for multiple IPSec groups. The ISAs are cold standby such that upon failure of the primary the standby resumes operation after the tunnels re-negotiate state. While the backup ISA can be shared by multiple tunnel groups only one tunnel group can fail to a single ISA at one time (no double failure support).

In multi-active mode, the active-**mda-number** value determines the number of ISA MDAs that will be active for this tunnel group, and tunnels are spread across all active ISA MDAs. Additional ISA MDA in this tunnel group will be in cold standby.

IPSec also supports dead peer detection (DPD).

Note that BFD can be configured on the private tunnel interfaces associated with GRE tunnels and used by the OSPF, BGP or static routing that is configured inside the tunnel.

SR OS also supports multi-chassis IPSec redundancy, which provides 1:1 stateful protection against MS-ISA failure or chassis failure

Statistics Collection

Input and output octets and packets per service queue are used for billing end customers who are on a metered service plan. Since multiple tunnels can be configured per interface the statistics can include multiple tunnels. These can be viewed in the CLI and SNMP.

Reporting (syslog, traps) for authentication failures and other IPSec errors are supported, including errors during IKE processing for session setup and errors during encryption or decryption.

A session log indicates the sort of SA setup when there is a possible negotiation. This includes the setup time, teardown time, and negotiated parameters (such as encryption algorithm) as well as identifying the service a particular session is mapped to, and the user associated with the session.

Security

The MS-ISA module provides security utilities for IPSec-related service entities that are assigned to interfaces and SAPs. These entities (such as card, isa-tunnel module, and IES or VPRN services) must be enabled in order for the security services to process. The module only listens to requests for security services from configured remote endpoints. In the case of a VPN concentrator application, these remote endpoints could come from anywhere on the Internet. In the cases where a point-to-point tunnel is configured, the module listens only to messages from that endpoint.

GRE Tunnel Multicast Support

GRE tunnels support unicast and multicast IP packets as payload. From a multicast prospective, a GRE tunnel IP interface (associated with a private tunnel SAP) can be configured as an IGMP interface and/or as a PIM interface; MLD is not supported. The following multicast features are supported:

- IGMP versions 1, 2 and 3
- IGMP import policies
- IGMP host tracking
- Static IGMP membership
- Configurable IGMP timers
- IGMP SSM translation
- Multicast CAC
- Per-interface, per-protocol (IGMP/PIM) multicast group limits
- MVPN support (draft-rosen)
- MVPN support (BGP-MPLS)
- PIM-SM and SSM operation
- PIM BFD support
- Configurable PIM timers
- Configurable PIM priority
- PIM tracking support
- PIM ECMP (bandwidth or hash-based)
- Static multicast route

IPv6 over IPv4 GRE Tunnel

IPv6 payload packets can be delivered over an IPv4 GRE tunnel. In this scenario the two endpoints of the GRE tunnel have IPv4 addresses and the VPRN or IES SAP interface to the tunnel is an IPv6 only or dual-stack IPv4/IPv6 interface. IPv6 over IPv4 GRE tunneling allows IPv6 islands to be connected over an IPv4 only transport infrastructure.

In order to configure a tunnel to carry IPv6 payload the tunnel must be configured with at least one **dest-ip** that contains an IPv6 address (global unicast and/or link local). A tunnel can have up to 16 **dest-ip** addresses (IPv4 and IPv6 together). For a tunnel to come operationally up all the **dest-ip** addresses must be part of locally configured subnets (associated with the private tunnel interface).

In order to forward IPv6 traffic through a tunnel supporting IPv6 payload a dynamic routing protocol (such as BGP or OSPFv3) can be configured to run inside the tunnel (by associating the protocol with the private tunnel interface) or else an IPv6 static route next-hop equal to a **dest-ip** of the tunnel can be used.

Note: IPv6 payload packets larger than 1280 bytes (the minimum IPv6 MTU) and also larger than the configured **ip-mtu** value of the tunnel are always discarded. If the **icmp6-generation** and **packet-too-big** commands are configured under the tunnel, then ICMPv6 Packet-Too-Big messages are generated and sent back to the originating host when discards occur due to the private side IP MTU being exceeded.

IKEv2

IKEv2, defined in RFC 4306, *Internet Key Exchange (IKEv2) Protocol*, is the second version of the Internet Key Exchange Protocol. The main driver of IKEv2 is to simplify and optimize the IKEv1. An IKE_SA and a CHILD_SA could be created with only 4 IKEv2 messages exchanges. The 7750-SR supports IKEv2 with following features:

- Static lan-to-lan tunnel.
- Dynamic lan-to-lan tunnel. Remote-access tunnel.
- Pre-shared-key authentication, certificate authentication, EAP (Remote-access tunnel only).
- Liveness check.
- IKE_SA rekey.
- Child_SA rekey.

IKEv2 TS-List

Since R12.0R1, the system allows users to configure a ts-list per ipsec-gw, apply to both IKEv2 remote-access tunnels and LAN-to-LAN tunnels.

Each ts-list contains up to 32 entries. Each entry contains a local address range/subnet. In case of ipsec-gw, the ts-list represents the TSr payload. Each entry represent one TS inside TSr.

The address range/subnet between entries in the same ts-list are NOT allowed to overlap.

The system will perform address range narrowing for the received TSr as following:

- The system will compute the intersection list between received TSr address ranges and address ranges in the ts-list.

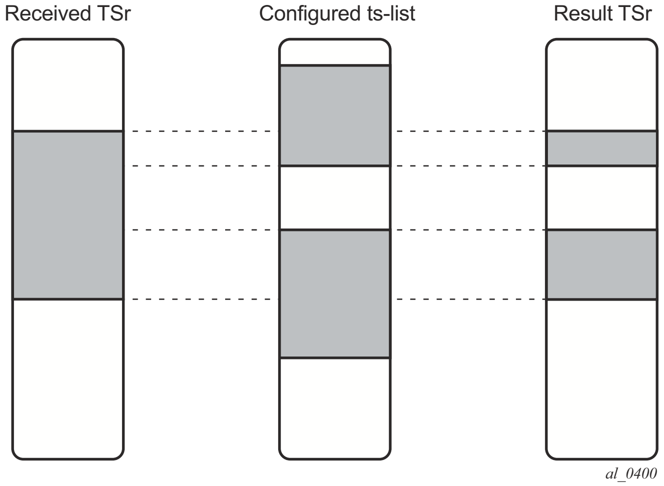

Figure 34: IKEv2 TS-List

- The system will send back the resulting TSr to the client.
- If there is no intersection, then system will fail the tunnel setup and return a TS_UNACCEPTABLE notification.

SHA2 Support

According to RFC 4868, *Using HMAC-SHA-256, HMAC-SHA-384, and HMAC-SHA-512 with IPsec*, the following SHA2 variants are supported for authentication or pseudo-random functions:

Use HMAC-SHA-256+ algorithms for data origin authentication and integrity verification in IKEv1/2, ESP:

- AUTH_HMAC_SHA2_256_128
- AUTH_HMAC_SHA2_384_192
- AUTH_HMAC_SHA2_512_256

For use of HMAC-SHA-256+ as a PRF in IKEv1/2:

- PRF_HMAC_SHA2_256
- PRF_HMAC_SHA2_384
- PRF_HMAC_SHA2_512

X.509v3 Certificate Overview

X.509v3 is an ITU-T standard which consists of a hierarchical system of Certificate Authorities (CAs) that issue certificates that bind a public key to particular entity's identification. The entity's identification could be a distinguished name or an alternative name such as FQDN or IP address.

An end entity is an entity that is not CA. For example an end entity can be a web server, a VPN client, or a VPN gateway.

A CA issues a certificate by signing an entity's public key with its own private key. A CA can issue certificates for an end entity as well as for another CA. In the case when a CA certificate is issued by itself (signed by its own private key), then this CA is called the root CA. Thus, an end entity's certificate could be issued by the root CA or by a subordinate CA (this is issued by another subordinate CA or root CA). When there are multiple CA involved, it is called a chain of CAs.

A PKI also includes the mechanism for revoking certificates due to reasons such as a compromised private key.

The certificate can be used for different purposes. One purpose is authentication. Typically certificate authentication functions as following:

- The system trusts a CA as trust anchor CA (which typically is a root CA). This means that all certificates issued by a trust anchor CA, or the certificates issued by a sub CA issued by the trust anchor CA, are consider trusted.
- A peer to be authenticated presents its certificate along with a signature over some shared data between the peer and system, which is signed by using a private key.
- The signature is verified by using the public key in the certificate. And the certificate itself is verified that is issued by the trust anchor CA or a sub-CA in a chain up to the trust anchor CA. The system can also check if the peer's certificate has been revoked. Only when all these verifications succeed, then the certificate authentication succeeds.

SROS X.509v3 Certificate Support

SROS's PKI implementation supports the following features:

- Certificate Enrollment:
  - → Locally generate RSA/DSA key
  - → Offline enrollment via PKCS#10
  - → Online enrollment via CMPv2
- Support CA chain
- Certificate revocation check:
  - → CRL for both EE (End Entity) and CA certificate

→ OCSP for EE certificate only

Local Storage

The SR OS requires the following objects to be stored locally as file:

- CA Certificate
- CRL
- System’s own certificate
- System’s own key

All above objects must be imported before they can be used by the SR OS. This is performed by using the **admin certificate import** command. The import process converts the format of input file to DER, encrypts the key file and saves it in cf3:/system-pki directory.

The imported file can also be exported as one to use in the specified format by means of the **admin certificate export** command.

The **admin certificate import** and **admin certificate export** command supports following formats:

- Certificates can be import/export by using following formats:
  - → PKCS#12
  - → PKCS#7 (DER and PEM)
  - → PEM
  - → DER

  Note: if there are multiple certificates in the file, only the 1st one will be used
- Key pair can be import/export by using following formats:
  - → PKCS#12 (must along with certificate)
  - → PEM
  - → DER
- CRL can be import/export by using following formats:
  - → PKCS#7 (DER and PEM)
  - → PEM
  - → DER
- PKCS#12 file can be encrypted with a password

CA-Profile

In SR OS, CA-related configuration is stored in a CA-profile which contains following configurations:

- Name and description
- CA's Certificate — An imported certificate
- CA's CRL— An imported CRL
- Revocation check method — Specifies the way CA check the revocation status of the certificate it issued.
- CMPv2 — A CMPv2 server related configurations
- OCSP— An OCSP responder related configurations

When user enables a ca-profile (no shutdown), system will load the specified CA certificate and CRL into memory. And following checks are performed:

- For CA certificate:
  - → All non-optional fields defined in section 4.1 of RFC 5280, *Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile*, must exist and conform to the RFC 5280 defined format.
  - → Check the version field to see if its value is 0x2.
  - → Check the Validity field to see that if the certificate is still in validity period.
  - → X509 Basic Constraints extension must exist and CA Boolean must be True.
  - → If Key Usage extension exists, then at least keyCertSign and cRLSign should be asserted.

For CRL:

- All non-optional fields defined in section 5.1 of RFC 5280 must exist and conform to the RFC 5280 defined format.
- If the version field exists, the value must be 0x1.
- The delta CRL Indicator must not exist (Delta CRL is not supported).
- CRL must be signed by the configured CA certificate.

CRL, by default, is required to enable ca-profile, but it could be optional by changing the revocation check method configuration. For the revocation check method configuration, refer to Certificate Revocation Check on page 439.

CA Chain Computation

In case of verifying a certificate with a CA or a chain of CAs, the system needs to identify the issuer CA of the certificate in question. The SR OS will look through all configured ca-profiles to find the issuer CA. The following is the method system used to find the issuer CA:

- The issuer CA's certificate subject must match the issuer field of the certificate in question.
- If present, the authority key identifier of the certificate in question must match the subject key identifier of the issuer CA's certificate
- If present, the key usage extension of the issuer CA's certificate must permit certificate signing.

Certificate Enrollment

The SR OS supports two certificate enrollment methods:

- Offline method via PKCS#10
- Online method via CMPv2

The offline method works as follows:

1. Generate a key pair via command "admin certificate gen-keypair"

   Example: admin certificate gen-keypair cf3:/segw.key size 2048 type rsa
2. Generate a PKCS#10 certificate signing request with the key generated in the step mentioned above via the **admin certificate gen-local-cert-req** command.

   Example: **admin certificate gen-local-cert-req keypair cf3:/segw.key subject-dn C=US,ST=CA,O=ALU,CN=SeGW domain-name segw-1.alu.com file cf3:/segw.pkcs10**

   Note: The user specifies the subject of certificate request and optionally can also specify a FQDN and/or an IP address as SubjectAltName.
3. Import the key file via the **admin certificate import** command

   Example: **admin certificate import type key input cf3:/segw.key output segw.key format der**
4. Since the key is imported, remove the key file generated in the first step for security reasons.
5. Send the PKCS#10 file to CA via an offline method such as E-MAIL.
6. CA signs the request, and returns the certificate.
7. Import the result certificate the **admin certificate import** command.

Example: **admin certificate import type cert input cf3:/segw.cert output segw.cert format pem**

For CMPv2-based enrollment, refer to Certificate Management Protocol Version 2 (CMPv2) on page 448.

Certificate Revocation Check

A revocation check is a process to see if a certificate has been revoked by the issuer CA.

The SR OS supports two methods for certificate revocation check:

- CRL
- OCSP

CRL can be used for both EE and CA certificate checks, while OCSP could only be used for an EE certificate.

With an IPSec application, users can configure multiple check methods with a priority order for an EE certificate. With the **status-verify** command in the **ipsec-tunnel/ipsec-gw configuration** context, a primary method, a secondary method and a default result can be configured. The primary and secondary method can be either OCSP or CRL. The default result is either **good** or **revoked**. If the system cannot get an answer from the primary method, then it will fall back to the secondary method. If secondary method also does not return an answer, then the system will use the default result.

By default, the system uses CRL to check the revocation status of a certificate, whether it is an end entity certificate or a CA certificate. This makes CRL a mandatory configuration in the ca-profile.

The **revocation-check** command in the **ca-profile** can change this behavior, with **revocation-check crl-optional** configured:

When a user enables the ca-profile (**no shutdown**), the system will try to load the configured CRL (specified by the **crl-file** command). But, if the system fails to load it for following reasons, then the system will still keep **ca-profile oper-up**, but treat the CRL as non-existent.

- The CRL file does not exist.
- The CRL is not properly encoded, possibly due to an interrupted file transfer.
- The CRL is not signed by the CA certificate configured in the CA profile.
- The wrong CRL version.
- The CRL expired or is not yet valid.

If the system needs to use the CRL of a specific **ca-profile** to check revocation status of an end entity certificate and CRL is non-existent due to the above reasons, then the system will treat it as unable to get an answer from CRL and fall back to the secondary status-verify method or default-result configured under the **ipsec-gw/ipsec-tunnel**.

If the system needs to check the revocation of a CA certificate in certificate chain, and if the CRL is non-existent due to the above reasons, then the system will skip checking the revocation status of the CA certificate. For example, the CA1 is issued by CA2, if CA2's **revocation-check** is **crl-optional** and CA2's CRL is non-existent, then the system will not check CA1 certificate's revocation status and consider it as good.

Note: The user must disable the **ca-profile** to change the revocation-check configuration.

For details about OCSP, refer to OCSP on page 449.

Certificate/CRL Expiration Warning

The system can optionally generate a warning message before a certificate or a CRL expires. The amount of time before expiration is configurable via two system-wide CLI commands (**certificate-expiration-warning** and **crl-expiration-warning**). The warning messages can also be optionally repeated at a configured interval. For details of the warning messages, refer to the corresponding command descriptions.

If a configured EE certificate expires, the system will not bring down an established ipsec-tunnel/ ipsec-gw down, however future certificate authentication will fail.

If a CA certificate expires, the system will bring the ca-profile operationally down. This will not affect established tunnels, however future certificate authentication that uses the ca-profile will fail.

Certificate/CRL/Key Cache

Configured certificates, CRLs, and keys are cached in memory before they are used by the system.

- Every certificate/CRL/Key has one cache copy system-wide.
- For a CA certificate and CRL, the cache will be created when there is a ca-profile and when a **no shutdown** is performed, and removed.
- For an ipsec-tunnel or ipsec-gw using legacy **cert** and **key** configurations, the cache will be created only when the first tunnel using it is in a **no shutdown** state, and it will be cleared when the last tunnel that used it is **shutdown**.
- For an ipsec-tunnel or ipsec-gw using **cert-profile**, the cache will be created when the first **cert-profile** using it is in a **no shutdown** state, and removed when the last cert-profile that used it is **shutdown.**

- If a certificate or key is configured with both a **cert-profile** and legacy **cert** or **key** command, then the cache will be created when the first object (a **ipsec-gw**, **ipsec-tunnel** or **cert-profile**) using it is in a **no shutdown** state and removed the last object using it is **shutdown**.

In order to update a certificate or key without a **shutdown ca-profile** or **ipsec-tunnel/ipsec-gw**, there is a CLI command (**admin certificate reload**) to manually reload the certificate and key cache. For details about reload, refer to the command description for **admin certificate reload**.

Auto CRL Update

The SR OS provides an automatic mechanism to update a CRL file. The system will try to download the CRL from a list of configured HTTP URLs and replace existing CRL file when a qualified CRL is successfully downloaded. A qualified CRL is a valid CRL signed by the CA and is more recent than the existing CRL. To determine if a downloaded CRL is more recent than an existing CRL, the system will compare the This-Update field of the CRL first. If they are the same, the system will compare the CRL number extension if present.

Note: The configured HTTP URL must point to a DER-encoded CRL file.

This features supports two types of downloading schedules:

- Periodic — The system will download a CRL periodically at the interval configured via the **periodic-update-interval** command. For example, if the **periodic-update-interval** is 1 day, then the system will download CRL every 1 day. The minimal periodic-update-interval is 1 hour.
- Next-update-based — The system will download a CRL at the time = Next_Update_time_of_current_CRL minus pre-update-time. For example, if the Nex-Update of current CRL is 2015-06-30 06:00 and pre-update-time is 1 hour, then the system will start the download at 2015-06-30, 05:00.

The system allows up to eight URLs to be configured for a given ca-profile. When downloading begins, URLs will be tried in order, and the first successfully downloaded qualified CRL will be used to update existing CRL. If the downloading fails or the downloaded CRL is not qualified, the system will move to the next URL in the list. If all URLs in the list fail to return a qualified URL, then:

- In case of next-update-based schedule, the system will wait for a configured retry-interval before retry from the first URL in the list again.
- In case of periodic schedule, the system will wait until the next scheduled time.

Upon executing a **no shutdown** of a ca-profile, if the auto-crl-update is enabled, then in case configures CRL file does not exist or is expired or invalid, then the system will start downloading right away.

The system also provides an **admin** command (**admin certificate crl-update ca** <*ca-profile-name*>) for users to manually trigger downloading. However, it requires a shutdown of the **auto-crl-update** command (**no auto-crl-update)**.

HTTP transport can be over either IPv4 or IPv6.

This feature support Base/Management/VPRN routing instance. VPLS management is not supported.In the case of VPRN, the HTTP server port can only be 80 or 8080.

Using Certificates For IPSec Tunnel Authentication

The SR OS supports X.509v3 certificate authentication for IKEv2 tunnel (LAN-to-LAN tunnel and remote-access tunnel). The SR OS also supports asymmetric authentication. This means the SR OS and the IKEv2 peer can use different methods to authenticate. For example, one side could use pre-shared-key and the other side could use a certificate.

The SR OS supports certificate chain verification. For a static LAN-to-LAN tunnel or ipsec-gw, there will be a configurable trust-anchor-profile which specifies the expecting CA(s) that should be present in the certificate chain before reaching the root CA (self-signed CA) configured in the system.

The SR OS's own key and certificate are also configurable per tunnel or ipsec-gw.

Note that when using certificate authentication, the SR OS will use the subject of the configured certificate as its ID by default.

Trust-Anchor-Profile

Since R12.0R1, the SR OS supports multiple trust-anchors per ipsec-tunnel/ipsec-gw. Users can configure a trust-anchor-profile that includes up to eight CAs. The system will build a certificate chain by using the certificate in the first certificate payload in the received IKEv2 message. If any of configured trust-anchor CAs in the trust-anchor-profile appears in the chain, then authentication is successful. Otherwise authentication is failed.

Note: The SR OS will only support processing of up to 16 hashes for the trust-anchor list from other products. If the remote end is sending more than 16, and a certificate match is in the > 16 range the tunnel will remain down with authentication failure.

The current **trust-anchor** command under ipsec-tunnel/ipsec-gw will be deprecated in a future release.

Cert-Profile

Since R12.0R1, the SR OS supports sending different certificate/chain according to the received IKEv2 certificate-request payload. This is achieved by configuring a cert-profile which allows up to eight entries. Each entry includes a certificate and a key and optionally also a chain of CA certificates.

The system will load cert/key in cert-profile into memory and build a chain: compare-chain for the certificate configured in each entry of cert-profile upon no shutdown of the cert-profile. These chains will be used in IKEv2 certificate authentication. If a chain computation cannot be completed for a configured certificate, then the corresponding compare-chain will be empty, or only partially computed.

Because there could be multiple entries configured in the cert-profile, the system needs to pick the cert/key in the correct entry that the other side expects to receive. This is achieved by a lookup of the CAs within the received cert-request payload in the compare-chain and then picking the first entry that there is a cert-request CA appearing in its chain. If there is no such cert, the system picks the first entry in the cert-profile. Note that the first entry is the 1st configured entry in cert-profile. The entry-id of first entry does not have to be “1”.

For example, there are three CA listed in certificate-request payload: CA-1, CA-2 and CA-3, and there are two entries configured in the cert-profile like following:

```
cert-profile “cert-profile-1”
     entry 1
          cert “cert-1”
          key “key-1”
     entry 2
          cert “cert-2”
          key “key-2”
          send-chain
               ca-profile “CA-1”
               ca-profile “CA-2”
```

The system will build two compare-chains: chain-1 for cert-1 and chain-2 for cert-2. Assume CA-2 appears in chain-2, but CA-1 and CA-3 do not appear in either chain-1 or chain-2. Then the system will pick entry 2.

After a cert-profile entry is selected, the system generates the AUTH payload by using the configured key in the selected entry. The system will also send the cert in the selected entry as “certificate” payload to the peer.

If a chain is configured in the selected entry, then one certificate payload is needed for each certificate in the configured chain. The first certificate payload in the IKEv2 message will be the signing certificate, which is configured by the **cert** command in the chosen cert-profile entry. With the above example, the system will send three certificate payloads: cert-2, CA-1,CA-2.

The following CA chain-related enhancements are supported

- The no-shut of a ca-profile will trigger a re-computation of compute-chain in related cert-profiles. The system will also generate a new log-1 to indicate a new compute-chain has been generated; the log includes the ca-profile names on the new chain. Another log-2 will be generated if the send-chain in a cert-profile entry is not in compute-chain due to this ca-profile change. Another log is generated if the hash calculation for a certificate under a ca-profile has changed.
- When no-shutting a cert-profile, the system now allows the CAs in the send-chain, not in the compute-chain. The system will also generate log-2 as above.
- The system now allows changes of the configuration of send-chain without shutdown of cert-profile.

Cert-Profile/trust-anchor-profile versus cert/trust-anchor

Since R12.0R1, cert-profile/trust-anchor-profile provides a superset of function of current **cert**/**trust-anchor** commands. The current **cert**/**trust-anchor** commands will be deprecated in a future release.

To facilitate transition and also to update the certificate trust-anchor, the following is a list of user configuration actions and corresponding system behavior while the tunnel or ipsec-gw is enabled (**no shutdown**):

- trust-anchor-profile **X** —> trust-anchor-profile **Y** : allowed
- trust-anchor **Z** —> trust-anchor-profile **Y** : allowed
- trust-anchor-profile **X** —> no trust-anchor-profile : disallowed
- trust anchor **W** —> trust-anchor **Z**: disallowed
- cert-profile **X** —> cert-profile **Y** : allowed
- cert A + key **B**—> cert-profile **Y** : allowed
- cert-profile **X** —> no cert-profile : disallowed
- cert **A** —> cert B : disallowed
- key **C** —> key D : disallowed

Notes:

- The new configuration will only be used in subsequent tunnel authentication. Existing tunnel will not be affected.
- The CLI rollback might not always allow above behavior.

Certificate Management Protocol Version 2 (CMPv2)

CMPv2, RFC 4210, *Internet X.509 Public Key Infrastructure Certificate Management Protocol* (*CMP*) is a protocol between a Certificate Authority (CA) and an end entity. It provides multiple certificate management functions like certificate enrollment, certificate update, etc.

The SR OS supports following CMPv2 operations:

- Initial Registration — The process the SR OS uses to enroll a certificate with a certain CA for the first time.
  - → Public/Private key pair must be pre-provisioned before enrollment by means of local generation or other methods.
  - → Users can optionally include a certificate or certificate chain in the extraCerts field of the initial registration request.
- Key Pair Update — A process for SR OS to update an existing certificate due to reason like refreshes key/cert before it expires or any other reason
- Certificate Update — A process where an initialized SR OS system obtains additional certificates.
- Polling — In some cases, the CA may not return the certificate immediately for reasons such as **request processing need manual intervention**. In such cases, the SR OS supports polling requests and responds as described in Section 5.3.22, Polling Request and Response, in RFC 4210, *Internet X.509 Public Key Infrastructure Certificate Management Protocol (CMP)*.

The following lists some implementation details:

- HTTP is the only supported transport protocol for CMPv2. HTTP 1.1 and 1.0 are supported and configurable.
- All CMPv2 messages sent by SR OS consist of only one PKI Message. The size of the sequence for PKI Messages are 1 in all cases.
- Both the password-based MAC and the public key-based signature CMPv2 message protection are supported.
- SR OS only allows one outstanding ir/cr/kur request for each CMPv2 server. The means that no new requests are allowed if a pending request is present.

OCSP

Online Certificate Status Protocol (OCSP) (RFC 2560, *X.509 Internet Public Key Infrastructure Online Certificate Status Protocol - OCSP*) is used by SR OS applications to determine the (revocation) state of an identified certificate. Unlike CRL, which relies on checking against an offline file, OCSP provides timely, online information regarding the revocation status of a certificate.

IPSec is the only supported application to use OCSP. With introduction of OCSP, the system supports both CRL and OCSP as the certificate revocation status checking method. For a given ipsec-tunnel or ipsec-gw, the user could configure a primary method, a secondary method and a default result to achieve a hierarchical fallback mechanism. If the primary method fails to return a result, the system will fall back to the secondary method. If the secondary method fails, the fall back proceeds to a default result.

The following lists implementation details:

- Only an OCSP client function is supported.
- HTTP is the only supported transport protocol.
- OCSP server access via management routing instance is not supported.
- SR OS does not sign an OCSP Request.
- The OCSP response must be signed. The system will verify the response by using the signer’s certificate included in the response. If there is no such certificate, the CA certificate in the ca-profile will be used.
- If a nextUpdate exists in the OCSP response, the system will check the current time <= nextUpdate. If yes, then the response is valid, otherwise the response is considered unreliable. The system will move to next revocation checking method.
- The revocation status result from a valid OCSP response will be cached in the system.
- OCSP can only be used to verify the revocation status of the end-entity certificate. CRL is still needed for CA certificate’s status verification.

Video Wholesale Example

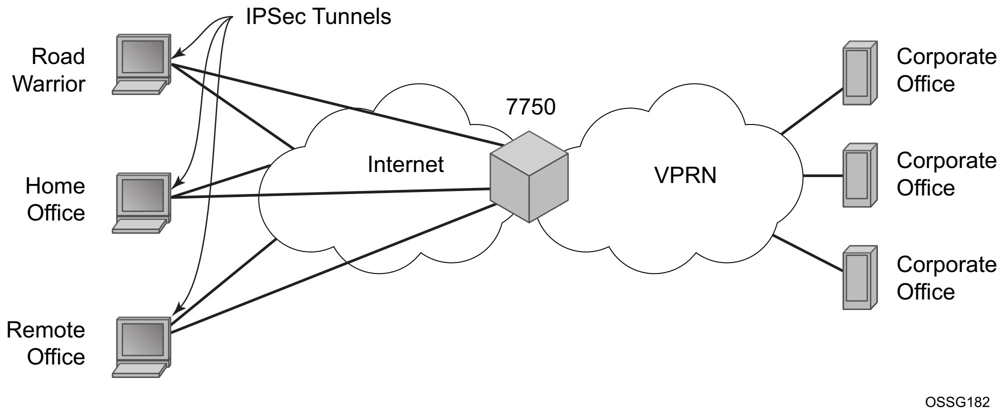

Figure 35: Video Wholesale Configuration

As satellite headend locations can be costly, many municipal and second tier operators cannot justify the investment in their own ground station in order to offer triple play features. However, it is possible for a larger provider or a cooperative of smaller providers to unite and provide a video headend. Each retail subscriber can purchase content from this single station, and receive it over IP. However, encryption is required so the signal cannot be understood if intercepted. A high speed encrypted tunnel is preferred over running two layers of double video protection which is cumbersome and computationally intensive.

Multi-Chassis IPSec Redundancy Overview

Multi-Chassis IPSec redundancy (MC-IPSec) provides a 1:1 (active/standby) IPSec stateful failover mechanism between two chassis.

- This feature provides protection against MS-ISA failure and chassis failure.
- IKEv2 static LAN-to-LAN is supported in SR OS R10.0R5; IKEv2 dynamic LAN-to-LAN tunnel is supported in SR OS R10.0R8. IKEv2 remote-access tunnel is supported in SR OS R11.0R6.
- This feature is supported on following platforms:
  - → 7750 SR7, SR12 and SR12E
  - → IOM3 and chassis mode D
  - → 7450 mixed mode
  - → Multi-active tunnel-group only
- The granularity of failover is per tunnel-group, which means a specific tunnel-group could failover to standby chassis independent of other tunnel-groups on the master chassis.
- The following components are included in this feature:
  - → Master Election: MIMP (MC-IPSec Mastership Protocol) runs between chassis to elect master, MIMP run for each tunnel-group independently
  - → Synchronization: MCS (Multi-Chassis Synchronization) sync IPSec states between chassis
  - → Routing:
    - – MC-IPSec aware routing attract traffic to the master chassis
    - – Shunting support
    - – MC-IPSec aware VRRP (10.0R8)

Architecture

The overall MC-IPSec redundancy architecture is displayed in Figure 36:

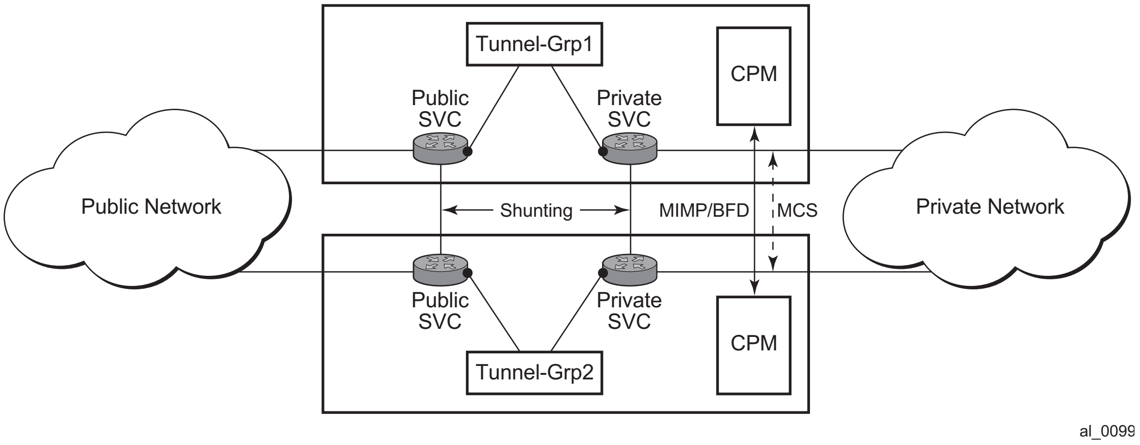

Figure 36: MC-IPSec Architecture

MC-IPSec Mastership Protocol (MIMP)

With MIMP enabled, there is a master chassis and a backup chassis. The state of the master or standby is per tunnel-group. For example (Table 15), chassis A and B, for tunnel-group 1, A is master, B is standby; for tunnel-group 2, A is standby, B is master.

Table 15: Master and Backup Chassis Example

	Master	Standby
Tunnel Group 1	A	B
Tunnel Group 2	B	A

All IKEv2 negotiation and ESP traffic encryption/decryption only occurs on the master chassis. If the backup chassis receives such traffic, if possible, it will shunt them to the master.

There will be a mastership election protocol (MIMP) running between the chassis to elect the master. This is an IP-based protocol to avoid any physical topology restrictions.

A central BFD session could be bound to MIMP to achieve fast chassis failure detection.

MIMP Protocol States

There are five MIMP states:

1. Discovery
   - Upon MC-IPSec is enabled for the tunnel-group, for example:
     - → System starts up.
     - → no shutdown MC-IPSec peer.
     - → no shutdown MC-IPSec tunnel-group.
   - Functionally, this means blackhole traffic to the MS-ISA and no shunting.
   - If the peer is reached before the discovery-interval (configurable) has expired, then the state will be changed to whatever the MIMP decides
   - If the peer is not reached before the discovery-interval has expired, then the state will be changed to **eligible** or **notEligible** depending on the oper-status of the tunnel-group.
2. notEligible
   - The tunnel-group is operationally down.
   - Functionally, this means blackhole traffic to the MS-ISA and no shunting.
3. Eligible
   - The peer is not reachable or the associated BFD session is down but the tunnel-group is operationally up.

- Functionally, this means the MS-ISA processes traffic.

4. Standby
    - Peer is reachable, elected standby.
    - Functionally, this means blackhole traffic to MS-ISA and shunting if possible.
5. Master
    - Peer is reachable, elected master.
    - Functionally, this means the MS-ISA processes traffic.

Election Logic

The following election logic is executed when MIMP packets are exchanged.

Calculate Master Eligibility:

1. Set masterEligible to TRUE if the local tunnel group is operationally up, otherwise FALSE.
2. Set peerMasterEligible to TRUE if the peer's tunnel group is operationally up, otherwise FALSE.

First elect based on eligibility:

3. If masterEligible and not peerMasterEligible, elect self master -> DONE.
4. If not masterEligible and peerMasterEligible, elect peer master -> DONE.
5. If not masterEligible and not peerMasterEligible, no master -> DONE.

Then apply stickyness rules (mastership tends not to change)

6. If l was "acting master" and peer was not "acting master", then elect self master -> DONE.
7. If 1 was not "acting master" and peer was "acting master", then elect peer master -> DONE.

Note: An "acting master" is either in MIMP state "master" or "eligible".

Then elect based on priority and number of active ISA:

8. If my priority is higher than peer, elect self master -> DONE.
9. If peer priority is higher than mine, elect peer master -> DONE.
10. If I have more active ISA than peer, elect self master -> DONE.
11. If peer has more active ISA than me, elect peer master -> DONE.

The tie breaker:

12. If the local chassis's MIMP source address is higher than the peer's, elect self master -> DONE.
13. Elect peer master -> DONE.

Protection Status

Each MC-IPSec-enabled tunnel-group has a "protection status", which could be one of following:

- notReady — The tunnel-group is not ready for a switchover due to reasons such as no elected standby to takeover or there are pending IPSec states which need to be synced. If switchover occurs with this status, then there could be a significant traffic impact.
- nominal — The tunnel-group is in a better situation to switchover than notReady. However, traffic still may be impacted.

Protection status serves as an indication for the operator to decide the optimal time to perform a controlled switchover.

The **show redundancy multi-chassis mc-ipsec peer** <*ip-address*> **tunnel-group** <*tunnel-group-id*>" command can be used to check current protection status.

Other Details

- Mastership election is per tunnel-group.
- MIMP is running in the base routing instance.
- MIMP will use the configured value of the **config>redundancy>multi-chassis>peer>source-address** command as the source address. If not configured, then system address will be used.
- The priority range is from 0 to 255.
- When an mc-ipsec tunnel-group enters standby from acting master, the tunnel-group will be restarted.
- When a tunnel-group enters an admin shutdown state under the mc-ipsec configuration (add a tunnel-group to mc-ipsec configuration, or upon admin shutdown of an mc-ipsec enabled tunnel group):
  → All tunnels in the tunnel-group will be deleted/reinstalled to the MS-ISAs.
  → All IKE states associated with those tunnels are locally purged from the MS-ISAs.
  → No IKE messages are sent to the IKE peer.

  These behaviors occur regardless of the presence of a redundant chassis or the state of a redundant chassis.

- With MC-IPSec enabled:
  - → auto-establish is blocked.
  - → For DPD configuration, only **no dpd** and **dpd** configurations with **reply-only** are allowed.

Routing

Routing in Public Service

A /32 route of the local tunnel address is created automatically for all tunnels on the MC-IPSec enabled tunnel-group.

This /32 route can be exported to a routing protocol by a route policy. The protocol type in route-policy is IPSec.

To attract traffic to the master chassis, a route metric of these /32 routes could be set according to the MIMP state, a metric from the master chassis is better than a metric from the standby chassis. There are three available states that can be used in the **from state** command in the route policy entry configuration:

- IPSec-master-with-peer
  → Corresponding MIMP states: master
- IPSec-master-without-peer
  → Corresponding MIMP states: eligible
- IPSec-non-master
  → Corresponding MIMP states: discovery/standby

However, if the standby chassis receives IPSec traffic, the traffic will be shunt to the master chassis by forwarding to a redundant next-hop. The redundant next-hop is an IP next-hop in the public routing instance.

Routing in Private Services

For static LAN-to-LAN tunnels, the static route with the IPSec tunnel as the next-hop could be exported to a routing protocol by a route policy. The protocol type remains **static**. For dynamic LAN-to-LAN tunnels, the reverse-route could be exported to a routing protocol by a route policy. The protocol type is **ipsec**. For remote-access tunnel, the private interface route could be exported to a routing protocol by a route policy.

Similar to routing in public services, the route metric of the above the routes could be set according to the MIMP state. Only a static route with an IPSec tunnel as the nexthop and reverse route has an MIMP state.

If the standby chassis receives IPSec traffic, the traffic will be shunt to the master chassis by forwarding to a redundant next-hop. The redundant next-hop is an IP next-hop in a private routing instance.

Other Details About Shunting

Shunting only works when tunnel-group is operationally up.

Shunting is not supported over auto-bind tunnels.

MC-IPSec Aware VRRP

In many cases, the public side is a Layer 2 network and VRRP is used to provide link or node protection. However, VRRP and MC-IPSec are two independent processes, each has its own mastership state, which means the VRRP master could be different from MC-IPSec master. This will result unnecessary shunting traffic.

To address this issue, MC-IPSec aware VRRP is introduced in SR OS Release 10.0R8, which add a new priority event in vrrp-policy: mc-ipsec-non-forwarding. If the configured tunnel-group enters non-forwarding (non-master) state, then the priority of associated VRRP instance will be set to the configured value. Delta priority is not supported for this type of event.

Synchronization

In order to achieve stateful failover, IPSec states are synced between chassis by using the MCS protocol.

- Only successfully created SA after a completed INITIAL EXCHANGES or CREATE_CILD_SA EXECHANGES is synced.
- Upon switchover, the new standby chassis will reboot the tunnel-group.
- The ESP sequence number is not synced.
- The CLI configuration is not synced.

The time must be the same on both chassis (using NTP/SNTP to sync to the same server is an option).

Automatic CHILD_SA Rekey

Because the ESP sequence number is not synced, a CHILD_SA rekey for each tunnel will be initiated by the new master to reset the sequence number upon switchover.

Responder Only

With MC-IPSec, it is required that MC-IPSec pair could only act as IKEv2 responder (except for the automatic CHILD_SA rekey upon switchover). To enable this behavior, configure following command.

```
config>isa>tunnel-grp>
     ipsec-responder-only
```

Refer to IPSec Deployment Requirements on page 460 section for details

IPSec Deployment Requirements

The following describes requirements to deploy SR OS IPSec features.

IPSec General:

To avoid high CPU loads and some complex cases, the following are the requirements for configuring IKEv2 lifetime:

1. The IKE_SA lifetime on one side should be approximately 2 times larger than the other side. he CHILD_SA lifetime on one side should be approximately 2 or 3 times larger than the other side.
2. With the previous rule, the lifetime of the side with smaller lifetime should NOT be too small:
   - IKE_SA: >= 86400 seconds
   - CHILD_SA: >= 3600 seconds
3. With 1st rule, on the side with smaller lifetime, the IKE_SA lifetime should be at least 3 times larger than CHILD_SA lifetime.
4. The IKE protocol is the control plane of IPSec, thus, the IKE packet should be treated as high QoS priority in the end-to-end path of public service.
   - On a public interface, a sap-ingress QoS policy should be configured to ensure the IKE packet treated as high QoS priority.
5. Correct system time is required for certificate authentication to work properly.

MC-IPSec Specific:

1. The IKEv2 lifetime requirements from the previous **IPSec General** section should be applied with special care to MC-IPSec deployments.

   In an MC-IPSec deployment where the MC-IPSec pair peers with single, non-redundant IKE clients, the IKEv2 lifetime requirements must be applied with the larger lifetimes configured on the MC-IPSec pair.

   An MC-IPSec deployment where one MC-IPSec pair peers with another MC-IPSec pair is not recommended. MC-IPSec performs optimally when the multi-chassis pair peers with a single IKE entity. If such a peering (MC-to-MC) is created, the above IKEv2 lifetime requirements should still be followed. However, with one side nominated to be the primary rekey initiator and having the smaller configured lifetimes.
2. Responder-only configuration is a mandatory requirement for all types of tunnels on the MC-IPSec pair in the usual deployment scenario of a MC-IPSec pair peering with single , non-redundant IKE clients.
3. DPD on the peer side, **dpd interval 300 max-retries 3 reply-only** on the MC-IPSec side.
4. Dedicated, direct physical link between chassis with enough bandwidth for MCS and shunting traffic.

MIMP/MCS and BFD for MC-IPSec traffic must be forwarded over resilient links so that a single IOM/IMM, MDA or port failure will not cause the MIMP to go down. Since this control traffic is forwarded in the base routing instance, the base routing instance links need to spread over multiple ports on multiple IOM/IMMs. Proper QoS configuration is needed to make sure the control traffic gets the highest priority.

5. A MC-IPSec switchover when the protection status is not nominal may result in unexpected behavior and traffic loss. A nominal state must be reached on both MC-IPSec chassis before a MC-IPSec switchover is triggered.
6. When using VRRP in the public service and a chassis failure occurs, the VRRP/Layer 2 network should re-converge before the MC-IPSec switchover occurs. One way to speed up VRRP switchover is to bind a BFD session to VRRP.
7. The system time of the master and standby chassis must be the same. One way to achieve this is for both chassis to sync to an NTP/SNTP server.

IKEv2 Remote-Access Tunnel

Since 11.0R6, SROS supports IKEv2 remote-access tunnel, the difference between a remote-access tunnel and LAN-to-LAN tunnel is remote-access tunnel allows client to request an internal address (and other attributes like DNS address) via IKEv2 configuration payload. The SR OS supports IKEv2 remote-access tunnel with following features:

- Authentication Methods:
  - → Pre-Shared-Key with RADIUS (**psk-radius**) or without RADIUS (**psk**)
  - → Certificate with RADIUS (**cert-radius**) or without RADIUS (**cert**)
  - → EAP/EAP-Only with RADIUS
- Internal address assignment via IKEv2 configuration payload
- Address assignment support:
  - → RADIUS server based
  - → Local Address assignment
- RADIUS accounting to report address usage
- RADIUS disconnect message to remove tunnel
- NAT-Traversal support
- Support MC-IPSec

Note: The SR OS only supports address assignments in first CHILD_SA negotiation.

IKEv2 Remote Access Tunnel – RADIUS-Based PSK/Certificate Authentication

If the **auth-method** parameter in the **ike-policy** is configured as **psk-radius** or **cert-radius**, then the system will authenticate the client via PSK or certificate accordingly as like a LAN-to-LAN tunnel. The difference being that in the case of **psk-radius** or **cert-radius,** the system will also perform a RADIUS authentication or authorization and optionally send RADIUS accounting messages.

Figure 37 displays a typical call flow for psk-radius and cert-radius.

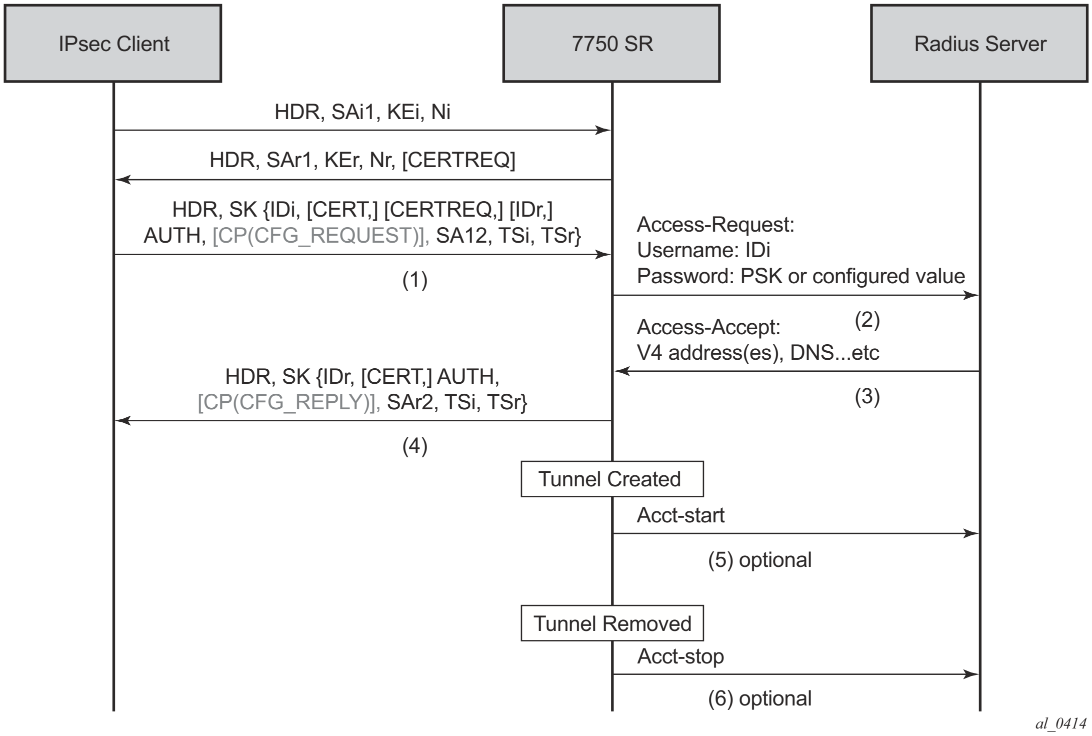
IPsec Client
7750 SR
Radius Server
HDR, SAi1, KEi, Ni
HDR, SAr1, KEr, Nr, [CERTREQ]
HDR, SK {IDi, [CERT,] [CERTREQ,] [IDr,]
AUTH, [CP(CFG_REQUEST)], SA12, TSi, TSr}
(1)
Access-Request:
Username: IDi
Password: PSK or configured value
(2)
Access-Accept:
V4 address(es), DNS...etc
(3)
HDR, SK {IDr, [CERT,] AUTH,
[CP(CFG_REPLY)], SAr2, TSi, TSr}
(4)
Tunnel Created
Acct-start
(5) optional
Tunnel Removed
Acct-stop
(6) optional
al_0414

Figure 37: Call Flow for psk-radius/cert-radius

The Access-Request includes following attributes:

- Username: IDi
- User-Password: One of following value’s hash according to section 5.2 of RFC 2865, *Remote Authentication Dial In User Service (RADIUS)*.
  - → Client’s PSK if the psk-radius is configured (refer to the CLI).
  - → Otherwise, a CLI configured key via the **password** command in the radius-authentication-policy; if password is not configured in this case, then system will not include User-Password attribute in access-request.
- Acct-Session-Id — Represents the IPSec tunnel session.
  The format is: local_gw_ip-remote_ip:remote_port-time_stamp.
  For example: 172.16.100.1-192.168.5.100:500-1365016423.
- Other RADIUS attributes (dependent on the **config>ipsec>radius-auth-policy> include-radius-attribute** configuration).
  - → Called-Station-Id: Local tunnel address.
  - → Calling-station-Id: Remote tunnel address:port number.
  - → Nas-Identifier: The system name.
  - → Nas-Ip-Address: The system IP.

→ Nas-port-id: The public tunnel SAP ID.

If the RAIDIUS authentication is successful, then the RADIUS sever will an send access-accept message back; otherwise, an access-reject message is sent back.

- The following are supported attributes in access-accept:
- Alc-IPsec-Serv-Id
- Alc-IPsec-Interface
- Framed-IP-Address
- Framed-IP-Netmask
- Alc-Primary-Dns
- Alc-Secondary-Dns
- Alc-IPsec-Tunnel-Template-Id
- Alc-IPsec-SA-Lifetime
- Alc-IPsec-SA-PFS-Group
- Alc-IPsec-SA-Encr-Algorithm
- Alc-IPsec-SA-Auth-Algorithm
- Alc-IPsec-SA-Replay-Window

Once the tunnel is successfully created, the system could optionally (depending on the configuration of the **radius-accounting-policy** under the **ipsec-gw** context), send an accounting-start packet to the RADIUS server, and also send an accounting-stop when the tunnel is removed. The user can also enable the **interim-update** option in the **radius-accounting-policy**.

The following are some attributes included in the acct-start/stop and interim-update:

- Acct-status-type
- Acct-session-id — The same as in the access-request
- Username

The following attributes are dependent on the **radius-acct-policy> include-radius-attribute** configuration:

- Frame-ip-address: the assigned internal address
- Calling-station-id
- Called-station-id
- Nas-Port-Id
- Nas-Ip-Addr
- Nas-Identifier
- Acct-Session-Time: tunnel session time, only in acct-stop packet.

Note: For a complete list of supported attributes, refer to the SR OS RADIUS Attributed Reference Guide.

The system also supports RADIUS disconnect messages to remove an established tunnel, If **accept-coa** (existing command) is enabled in the radius-server configuration, then the system will accept the disconnect-request message (RFC 5176, *Dynamic Authorization Extensions to Remote Authentication Dial In User Service* (*RADIUS*)), and tear down the specified remote-access tunnel.

```
config>router>radius-server>server#
     [no] accept-coa
```

Note: For security reasons, the system will only accept a disconnect-request when **accept-coa** is configured **and** the disconnect-request comes from the corresponding server.

The target tunnel is identified by one of following methods:

- Acct-Session-Id
- Nas-Port-Id + Framed-Ip-Addr(Framed-Ipv6-Prefix) + Alc-IPsec-Serv-Id
- User-Name

Refer to the SROS RADIUS Attribute Reference Guide for more details about disconnect message support.

By default, the system will only return what the client has requested in the CFG_REQUEST payload. However, this behavior can be overridden by configuring **relay-unsolicited-cfg-attribute** in the **ike-policy**. With this configuration, the configured attributes returned from the source (such as the RADIUS server) will be returned to the client regardless if the client has requested it in the CFG_REQUEST payload.

IKEv2 Remote-Access Tunnel – EAP Authentication

The SR OS supports EAP authentication for a IKEv2 remote-access tunnel, in which case, the system acts as an authenticator between an IPSec client and a RADIUS server. It transparently forwards EAP messages between the IKEv2 session and RADIUS session. Thus, the actual EAP authentication occurs between the client and the RADIUS server.

Figure 38 shows a typical call flow of EAP authentication:

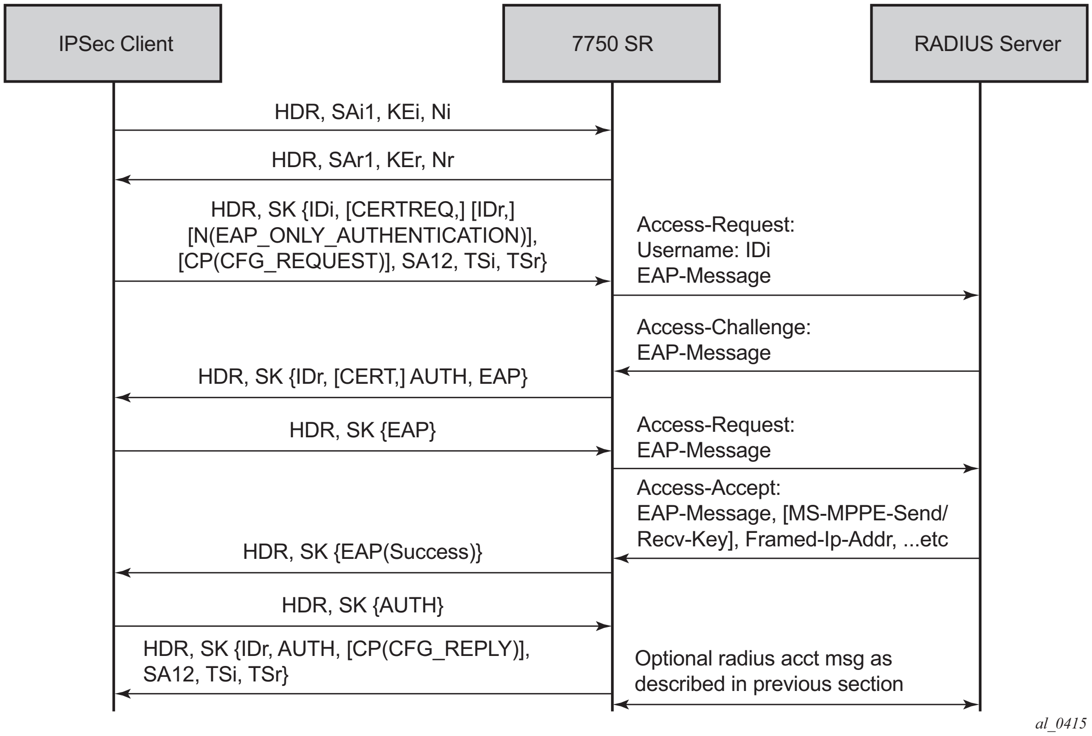
IPSec Client
7750 SR
RADIUS Server
HDR, SAi1, KEi, Ni
HDR, SAr1, KEr, Nr
HDR, SK {IDi, [CERTREQ,] [IDr,]
[N(EAP_ONLY_AUTHENTICATION)],
[CP(CFG_REQUEST)], SA12, TSi, TSr}
Access-Request:
Username: IDi
EAP-Message
Access-Challenge:
EAP-Message
HDR, SK {IDr, [CERT,] AUTH, EAP}
HDR, SK {EAP}
Access-Request:
EAP-Message
Access-Accept:
EAP-Message, [MS-MPPE-Send/
Recv-Key], Framed-Ip-Addr, ...etc
HDR, SK {EAP(Success)}
HDR, SK {AUTH}
HDR, SK {IDr, AUTH, [CP(CFG_REPLY)],
SA12, TSi, TSr}
Optional radius acct msg as
described in previous section
al_0415

Figure 38: Typical Call Flow of EAP Authentication

EAP authentication is enabled by configuring **authentication eap**. Once enabled, after the received IKE_AUTH request from the client, the system sends an EAP-Response/ID with IDi as the value in the access-request to AAA. AAA will return a method request and the system starts passing through between the client and AAA. (as shown in Figure 38).

The generation of the AUTH payload in the IKE_AUTH response sent by the SR OS (message 4 in flow shown above) is dependent on the **own-auth-method** configuration:

- psk — The AUTH payload is present and generated by using PSK.
- cert — The AUTH payload is present and generated by the configured public and private key pairs as it does in certificate authentication. Any needed certificates will be also sent.

- eap-only — Neither AUTH nor CERT payload is present.

The RADIUS attributes in authentication and accounting packets are similar as psk-radius and cert-radius with following differences:

- RADIUS attributes support EAP-Message/Message-Authenticator /State attributes
- RADIUS attributes support Access-Challenge packet
- RADIUS attributes support MS-MPPE-Send-Key/ MS-MPPE-Recv-Key in access-accept. These two attributes are required for all EAP methods that generate MSK.

The system provides a method to support EAP and other authentication methods on the same **ipsec-gw** policy. This is enabled by configuring **auto-eap-radius** or **auto-eap** as the **auth-method** in the **ike-policy**.

With **auto-eap-radius**:

- If there is no AUTH payload in an IKE_AUTH request, then the system uses EAP to authenticate the client and will also use **own-auth-method** to generate the AUTH payload.
    - → If there is an AUTH payload in the IKE_AUTH request:
    - → If the **auto-eap-method** is **psk**, then the system proceeds as auth-method: psk-radius.
    - → If the **auto-eap-method** is **cert**, then the system proceeds as auth-method: cert-radius.
    - → If **auto-eap-method** is **psk-or-cert**, then:
        - – If the Auth Method field of the AUTH payload is PSK, then the system proceeds as **auth-method:psk-radius**.
        - – If the Auth Method field of the AUTH payload is RSA or DSS, then the system proceeds as **auth-method:cert-radius**.

The system will use **auto-eap-own-method** to generate the AUTH payload.

With auto-eap:

- If there is no AUTH payload in IKE_AUTH request, then the system uses EAP to authenticate the client and will also use **own-auth-method** to generate AUTH payload.
- If there is an AUTH payload in the IKE_AUTH request:
    - → If the **auto-eap-method** is **psk**, then the system proceeds as auth-method: psk.
    - → If the **auto-eap-method** is **cert**, then the system proceeds as auth-method: cert.
    - → If the **auto-eap-method** is **psk-or-cert**, then:
    - → If the Auth Method field of the AUTH payload is PSK, then the system proceeds as **auth-method psk**.
    - → If the Auth Method field of the AUTH payload is RSA or DSS, then the system proceeds as **auth-method cert-auth**

The system will use auto-eap-own-method to generate the AUTH payload.

IKEv2 Remote-Access Tunnel – Authentication without RADIUS

To achieve authentication without RADIUS, auth-method need to configured as psk or cert-auth and local address assignment must be configured under ipsec-gw.

Figure 39 shows a typical call flow of certificate or PSK authentication without RADIUS.

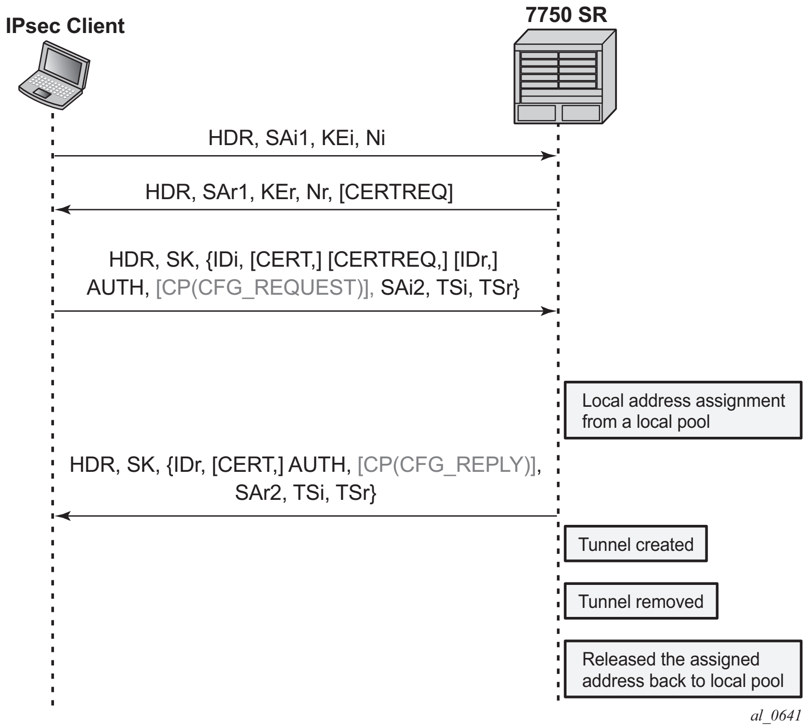
IPsec Client
7750 SR
HDR, SAi1, KEi, Ni
HDR, SAr1, KEr, Nr, [CERTREQ]
HDR, SK, {IDi, [CERT,] [CERTREQ,] [IDr,]
AUTH, [CP(CFG_REQUEST)], SAi2, TSi, TSr}
Local address assignment from a local pool
HDR, SK, {IDr, [CERT,] AUTH, [CP(CFG_REPLY)],
SAr2, TSi, TSr}
Tunnel created
Tunnel removed
Released the assigned address back to local pool
al_0641

Figure 39: Typical Call Flow of Certificate or PSK Authentication without RADIUS

Figure 40 shows a typical call flow for EAP authentication.

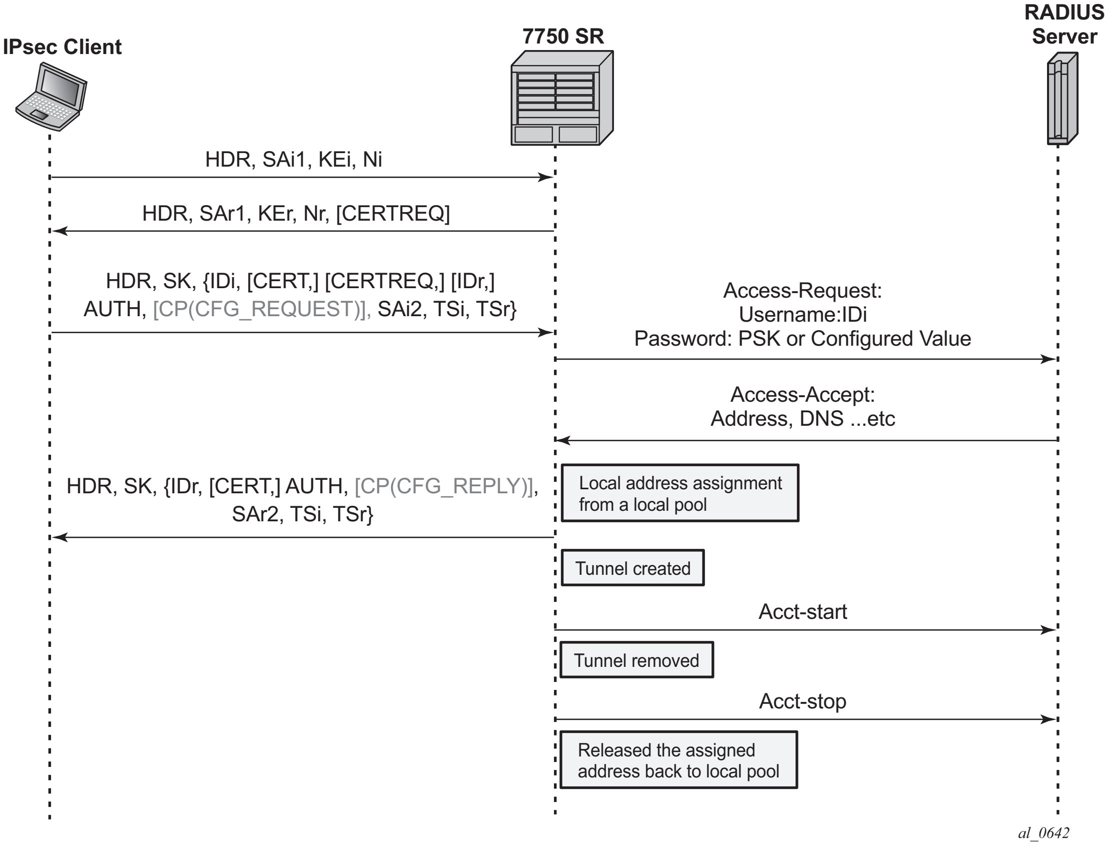
IPsec Client
7750 SR
RADIUS Server
HDR, SAi1, KEi, Ni
HDR, SAr1, KEr, Nr, [CERTREQ]
HDR, SK, {IDi, [CERT,] [CERTREQ,] [IDr,] AUTH, [CP(CFG_REQUEST)], SAi2, TSi, TSr}
Access-Request: Username:IDi Password: PSK or Configured Value
Access-Accept: Address, DNS ...etc
HDR, SK, {IDr, [CERT,] AUTH, [CP(CFG_REPLY)], SAr2, TSi, TSr}
Local address assignment from a local pool
Tunnel created
Acct-start
Tunnel removed
Acct-stop
Released the assigned address back to local pool
al_0642

Figure 40: Typical Call Flow for EAP Authentication

In this configuration, the **radius-authentication-policy** and **radius-accounting-policy** in the **ipsec-gw** context are ignored.

RADIUS disconnect messages are supported in this case. Only the following tunnel identification methods are supported:

- Nas-Port-Id + Framed-Ip-Addr(Framed-Ipv6-Prefix) + Alc-IPsec-Serv-Id
- User-Name

IKEv2 Remote-Access Tunnel – Address Assignment

The SROS supports the following methods of address assignment for IKEv2 remote-access tunnels:

- RADIUS based
- Local address assignment (LAA)
- DHCP based

In case of RADIUS-based address assignments, the address information is returned in access-accept packet. This implies that the RADIUS-based address assignment requires using an auth-method with RADIUS, such as psk-radius, cert-radius, eap.

In case of LAA, the system gets an address from a pool defined in a local DHCPv4/v6 server. When a tunnel is removed, the assigned address is released back to the pool. If the local DHCPv4/v6 server is shutdown, then all existing tunnels that have an address from the server will be removed. If LAA is shutdown, the current established tunnel that used LAA will stay up.

In case of DHCP-based, the system acts as a DHCP client on behalf of the IPSec client and requests an address from an external DHCP server via standard DHCP exchange. In this case, the system also acts as a DHCP relay agent which relays all DHCP packets between the DHCP server and the local DHCP client. DHCP renew and rebind are also supported.

The chaddr in DHCPv4 header is generated by SROS:

- First two bytes of the MAC address are 02:03
- The rest of 4 bytes are the hash result of IKEv2 IDi

The following options are included in DHCPv4 packet sent by SROS:

- Option 82 Circuit-id: <private-SAP-id>|<private-interface-name>; e.g. tunnel-1.private:100|priv-int
- Option 82 remote-id:IKEv2 IDi in text format
- Option 61 Client-Id: one byte that represents the IKEv2 IDi type + IKEv2 IDi in text format. The value of first byte is following:
  → ID_IPV4_ADDR = 1
  → ID_DER_ASN1_DN = 2
  → ID_FQDN = 3
  → ID_RFC822_ADDR = 4
  → ID_IPV6_ADDR = 5

Notes:

- Using a local DHCP server of same chassis for DHCP-based address assignment is not supported. The DHCP server must be external.
- replay-proxy (**config>service>vprn>if>dhcp>relay-proxy**) on interface that has gi-address as the interface address must be enabled to use a DHCPv4 address assignment.
  - → In case that the DHCP server resides in a private service, and the gi-address is an address configured on the corresponding tunnel interface, then the relay-proxy must be enabled on the corresponding private interface.
  - → In case that the DHCP sever resides in a routing instance X that is different from private service, then there must be an interface (such as a loopback interface) M in X that has the gi-address as the interface address and gi-address must be routable for the DHCP server. Also, the relay-proxy must be enabled on interface M.

The biggest difference between LAA and DHCP-based method is LAA uses a local API to get an address from a local pool. There is no DHCP packet exchange in case of LAA while a DHCP-based method uses standard DHCP packet exchange to request a packet from an external DHCP server.

Since there are three methods for address assignment, the following is the priority order (descending) to which source to choose in case that more than one source is configured:

1. LAA
2. DHCP
3. RADIUS

LAA/DHCP could work with an auth-method that does not involve RADIUS, as well as an auth-method that involves RADIUS. In the case of using LAA/DHCP with RADIUS involving auth-method:

- The address information returned by RADIUS server will be ignored (even if LAA/DHCP is configured but is shutdown).
- Non-address-related attributes in access-accept such as Alc-IPsec-Serv-Id, Alc-IPsec-Tunnel-Template-Id, etc., will still be accepted.
- RADIUS accounting is supported in this case, but the Framed-IP-Addr/Framed-IPv6-Prefix reported in acct-request packet is the LAA/DHCP assigned address, not the address returned by RADIUS server.
- RADIUS disconnect message is supported

In case MC-IPsec:

- LAA — The configuration of **config>redundancy>multi-chassis>peer >sync local-dhcp-server** is not needed. This is because the assigned address will be synced as part of the IPSec tunnel states.
- DHCP:
  - → The DHCP packet exchange process only occurs on the master
  - → The assigned address is synced to standby as part of the IPSec states. Standby will not initiate any DHCP exchanges.
  - → The DHCP server address (**ipsec-gw>dhcp>server**) configured should be the same on both chassis:
  - → After an MC switchover:
    - – The new master will not initiate any DHCP process unless it is time to renew an address or a tunnel goes down.
    - – If a new master needs to renew a address or release an address, it will send the DHCP packet to the same DHCP server address that assigned the address on the old master, assume the external DHCP server is still on, and then renew or release will be processed normally.
    - – If the new master needs to assign an address for a new tunnel setup, it will send a DHCP discovery to all configured DHCP server addresses and then pick the first offer to finish DORA process.
  - → For DHCPv4, a gi-address is used by server to forward a response back, so the gi-address must be an interface address of SR OS. In the case of multi-chassis, if a DHCPv4 server resides in a private VPRN, then there are two options:
    - – Configure the same private interface address on both chassis and then use it as gi-address. Configure MC-IPSec aware routing to make sure that the DHCP response is attracted to the master.
    - – Configure different private interface addresses (but with same subnet) on both chassis. The gi-address is the private interface address of local chassis. Beside the private subnet, two /32 private interface address route from two chassis will also need to be advertised so that the DHCP response is routed to the correct chassis.

If the DHCPv4 server does not reside in a private VPRN, then one method is to configure a loopback interface with a /32 address in the private subnet; and the loopback interface address will be used as the gi-address. Different address must be configured on the master and standby chassis.

IPv6 IPSec Support

The SROS provides the following IPv6 support to IPSec functions:

- IPv6 packets as the ESP tunnel payload
- IPv6 as the ESP tunnel encapsulation

IPv6 as Payload

IPv6 as payload allows IPv6 packets to be forwarded within an IPSec tunnel. Current support includes the following:

- Tunnel type support:
  - → Static LAN-to-LAN tunnel
  - → Dynamic LAN-to-LAN tunnel
  - → Remote-access tunnel (only IKEv2 is supported)
- The prefix length of the IPv6 address on a private interface must be /96 or longer

IPv6 as Payload: Static LAN-to-LAN Tunnel

There are three methods to forward IPv6 traffic into static tunnels on the private side:

1. The destination address is a configured destination IP (dest-ip) under the tunnel context:
   ```
   config>service>vprn>if>sap>ipsec-tun
        [no] dest-ip <v4/v6 addr>
   ```
   - The dest-ip can be either an IPv6 address or an IPv4 address.
   - In the case of IPv6, it must be either an IPv6 global unicast address or an IPv6 link-local address.
   - In the case of IPv4, it can be used to forward IPv4 traffic into the tunnel.
   - In case of unicast address, dest-ip must be within the prefix configured on the private interface.
   - Up to 16 destination IPs can be configured per ipsec-tunnel.
2. A v6 route with a configured destination IP as the next-hop, this route can be learned from either a static or dynamic from a routing protocol such as BGP.
3. An IPv6 static route with an ipsec-tunnel used as the next-hop.

A security policy supports either an IPv4 entry or an IPv6 entry or both for dual-stack.

IPv6 as Payload: Dynamic LAN-to-LAN Tunnel

With dynamic LAN-to-LAN tunnels, the system will automatically create a v6 reverse route in the private VPRN based on the received TSi payload with the tunnel as the nexthop.

IPv6 as Payload: Remote-Access Tunnel

The system supports the following IKEv2 IPv6 configuration attributes:

- INTERNAL_IP6_ADDRESS
- INTERNAL_IP6_DNS

The system supports only one internal IPv6 address per tunnel. The following IPv6-related RADIUS attributes are also supported in access-accept:

- Framed-IPv6-Prefix will be translated into INTERNAL_IP6_ADDRESS in the configuration payload, which includes two parts. A 16-byte v6 prefix and a one-byte prefix length.
- Alc-Ipv6-Primary-Dns
- Alc-Ipv6-Secondary-Dns

If an internal v6 address has been assigned to the remote-access client, then the Framed-IPv6-Prefix will also be included in RADIUS accounting-request packet.The assigned internal v6 address must be within the prefix configured on the corresponding private interface.

If the client request both v4 and v6 address and address source (such as RADIUS or LAA) assign both v4 and v6 address, then both v4 and v6 addresses will be assigned to the client via the configuration payload.

IPv6 as Encapsulation

IPv6 as encapsulation allows IPv4 or IPv6 packets to be forwarded within an IPv6 ESP tunnel, also the IKE protocol can run over IPv6. Current support includes:

- Tunnel type support:
  - → Static LAN-to-LAN tunnel
  - → Dynamic LAN-to-LAN tunnel
  - → Remote-access tunnel (For IKEv1, only v4 over v6 is supported)

For a given ipsec-gw or ipsec-tunnel, only one local gateway address is supported, which could be either an IPv4 or IPv6 address. The SR OS also provides fragmentation and reassembly support for IPv6 ESP/IKE packets.