

IPSec Network Applications

This chapter describes several methods for implementing IPSec within various network applications.

Topics discussed in this chapter include:

- Implementing IPSec for PDN Access Applications, on page 1
- Implementing IPSec for Mobile IP Applications, on page 3
- Implementing IPSec for L2TP Applications, on page 9
- IPSec for LTE/SAE Networks, on page 15
- IPSec for Femto-UMTS Networks, on page 24

Implementing IPSec for PDN Access Applications

This section provides information on the following topics:

- How IPSec-based PDN Access Configuration Works, on page 1
- Configuring IPSec Support for PDN Access, on page 2

This section assumes that ISAKMP crypto maps are used as opposed to manual crypto maps.

How IPSec-based PDN Access Configuration Works

The following figure and the text that follows describe how sessions accessing a PDN using IPSec are processed by StarOS.

Table 1: IPSec PDN Access Processing

Step	Description
1	A subscriber session or PDP context Request, in GGSN service, arrives at the system.
2	The system processes the subscriber session or request as it would typically.
3	Prior to routing the session packets, the system compares them against configured Access Control Lists (ACLs).

Step	Description
4	The system determines that the packet matches the criteria of an ACL that is associated with a configured crypto map.
5	From the crypto map, the system determines the following: • The map type, in this case ISAKMP • The pre-shared key used to initiate the Internet Key Exchange (IKE) and the IKE negotiation mode • The IP address of the security gateway • Whether perfect forward secrecy (PFS) should be enabled for the IPSec SA and if so, what group should be used • IPSec SA lifetime parameters • The name of a configured transform set defining the IPSec SA
6	To initiate the IKE SA negotiation, the system performs a Diffie-Hellman exchange of the pre-shared key specified in the crypto map with the specified peer security gateway.
7	The system and the security gateway negotiate an ISAKMP policy (IKE SA) to use to protect further communications.
8	Once the IKE SA has been negotiated, the system negotiates an IPSec SA with the security gateway using the transform method specified in the transform sets.
9	Once the IPSec SA has been negotiated, the system protects the data according to the IPSec SAs established during step 8 and sends it over the IPSec tunnel.

Configuring IPSec Support for PDN Access

This section provides a list of the steps required to configure IPSec functionality on the system in support of PDN access. Each step listed refers to a different section containing the specific instructions for completing the required procedure.

Important These instructions assume that the system was previously configured to support subscriber data sessions either as a core service or an HA. In addition, parameters configured using this procedure must be configured in the same destination context on the system.

Step 1 Configure one or more IP access control lists (ACLs) according to the information and instructions located in the *IP Access Control Lists* chapter of the product Administration Guide.

Step 2 Configure one or more transform sets according to the instructions located in the *Transform Set Configuration* chapter of this guide.

Step 3 Configure one or more ISAKMP policies according to the instructions located in the *ISAKMP Policy Configuration* chapter of this guide.

Step 4 Configure an ipsec-isakmp crypto map according to the instructions located in the *ISAKMP Crypto Map Configuration* section of the *Crypto Maps* chapter in this guide.

Step 5 Apply the crypto map to an interface on the system according to the instructions located in the *Crypto Map and Interface Association* section of the *Crypto Maps* chapter in this guide.

Step 6 Save your configuration to flash memory, an external memory device, and/or a network location using the Exec mode command **save configuration**. For additional information on how to verify and save configuration files, refer to the *System Administration Guide* and the *Command Line Interface Reference*.

Implementing IPSec for Mobile IP Applications

This section provides information on the following topics:

- How IPSec-based Mobile IP Configuration Works, on page 3
- Configuring IPSec Support for Mobile IP, on page 7
- RADIUS Attributes for IPSec-based Mobile IP Applications, on page 8

How IPSec-based Mobile IP Configuration Works

The following figure and the text that follows describe how Mobile IP sessions using IPSec are processed by the system.

Figure 1: IPSec-based Mobile IP Session Processing

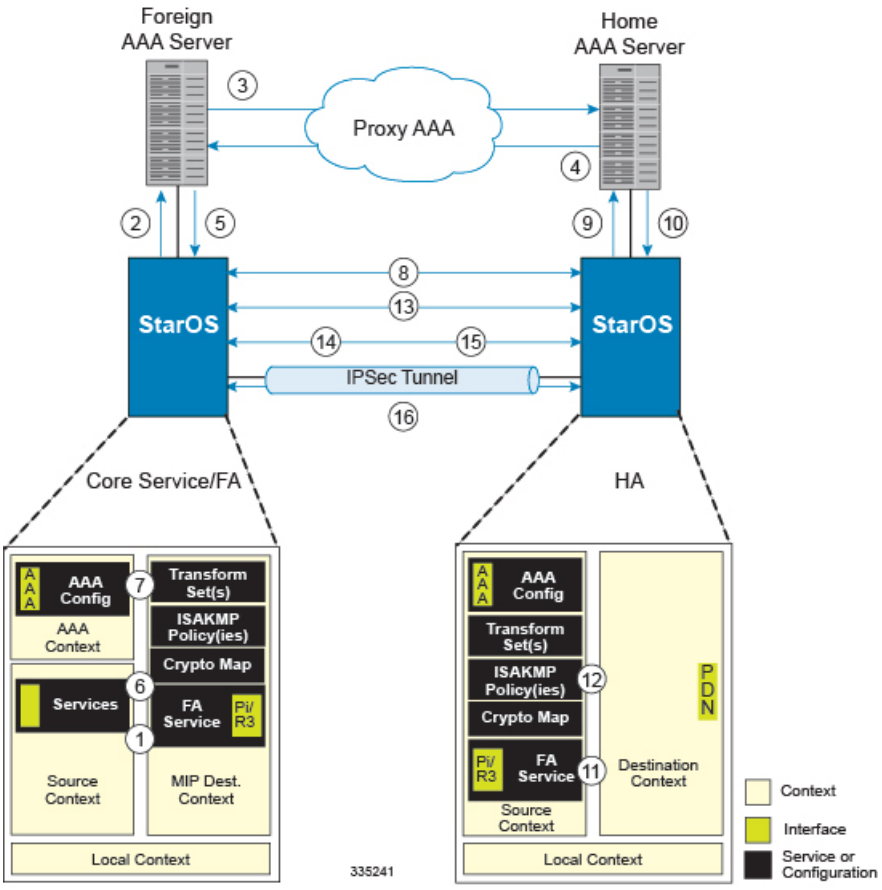


Table 2: IPSec-based Mobile IP Session Processing 0

Step	Description
1	FA service receives a Mobile IP registration request from the mobile node.
2	FA sends an Access-Request to the FAAA server with the 3GPP2-IKE-Secret-Request attribute equal to yes.
3	The FAAA proxies the request to the HAAA.

Step	Description
4	The HAAA returns an Access-Accept message including the following attributes: • 3GPP2-Security-Level set to 3 for IPSec tunnels and registration messages • 3GPP2-MIP-HA-Address indicating the IP address of the HA with which the FA is to communicate • 3GPP2-KeyId providing an identification number for the IKE secret (alternatively, the keys may be statically configured for the FA and/or HA) • 3GPP2-IKE-Secret indicating the pre-shared secret to use to negotiate the IKE SA
5	The FAAA passes the accept message to the FA with all of the attributes.
6	The FA determines if an IPSec SA already exists based on the HA address supplied. If so, that SA will be used. If not, a new IPSec SA will be negotiated.
7	The FA determines the appropriate crypto map to use for IPSec protection based on the HA address attribute. It does this by comparing the address received to those configured using the isakmp peer-ha command. From the crypto map, the system determines the following: • The map type, in this case dynamic • Whether perfect forward secrecy (PFS) should be enabled for the IPSec SA and if so, what group should be used • IPSec SA lifetime parameters • The name of one or more configured transform set defining the IPSec SA
8	To initiate the IKE SA negotiation, the FA performs a Diffie-Hellman (D-H) exchange of the ISAKMP secret specified in the IKE secret attribute with the peer HA dictated by the HA address attribute. Included in the exchange is the Key ID received from the HAAA.

Step	Description
9	Upon receiving the exchange, the HA sends an access request to the HAAA with the following attributes: • 3GPP2-S-Request (note that this attribute is not used if the IPSec keys are statically configured) • 3GPP2-User-name (the username specified is the IP addresses of the FA and HA). The password used in the access request is the RADIUS shared secret.
10	The HAAA returns an Access-Accept message to the HA with the following attributes: • 3GPP2-S indicating the "S" secret used to generate the HA's response to the D-H exchange • 3GPP2-S-Lifetime indicating the length of time that the "S" secret is valid • 3GPP2-Security-Level set to 3 for IPSec tunnels and registration messages (optional)
11	The HA determines the appropriate crypto map to use for IPSec protection based on the FA's address. It does this by comparing the address received to those configured using the isakmp peer-fa command. From the crypto map, the system determines the following: • The map type, in this case dynamic • Whether perfect forward secrecy (PFS) should be enabled for the IPSec SA and if so, what group should be used • IPSec SA lifetime parameters • The name of one or more configured transform set defining the IPSec SA
12	The HA creates a response to the D-H exchange using the "S" secret and the Key ID sent by the FA.
13	The HA sends IKE SA negotiation D-H exchange response to the FA.
14	The FA and the HA negotiate an ISAKMP (IKE) policy to use to protect further communications.

Step	Description
15	Once the IKE SA has been negotiated, the system negotiates an IPSec SA with the security gateway using the transform method specified in the transform sets.
16	Once the IPSec SA has been negotiated, the system protects the data according to the IPSec SAs established during step 15 and sends it over the IPSec tunnel.



Important Once an IPSec tunnel is established between an FA and HA for a particular subscriber, all new Mobile IP sessions using the same FA and HA are passed over the tunnel regardless of whether or not IPSec is supported for the new subscriber sessions. Data for existing Mobile IP sessions is unaffected Configuring IPSec

Configuring IPSec Support for Mobile IP

This section provides a list of the steps required to configure IPSec functionality on the system in support of Mobile IP. Each step listed refers to a different section containing the specific instructions for completing the required procedure.



Important These instructions assume that the systems were previously configured to support subscriber data sessions either as an FA or an HA.

Step 1 Configure one or more transform sets for the FA system according to the instructions located in the *Transform Set Configuration* chapter of this guide.

The transform set(s) must be configured in the same context as the FA service.

Step 2 Configure one or more ISAKMP policies or the FA system according to the instructions located in the *ISAKMP Policy Configuration* chapter of this guide.

The ISAKMP policy(ies) must be configured in the same context as the FA service.

Step 3 Configure an ipsec-isakmp crypto map or the FA system according to the instructions located in the *Dynamic Crypto Map Configuration* section of the *Crypto Maps* chapter of this guide.

The crypto map(s) must be configured in the same context as the FA service.

Step 4 Optional. Configure DPD for the FA to help prevent IPSec tunnel state mismatches between the FA and HA according to the instructions located in the *Dead Peer Detection (DPD) Configuration* section of the *Redundant IPSec Tunnel Fail-Over* chapter of this guide.

Important Though the use of DPD is optional, it is recommended in order to ensure service availability.

Step 5 Configure the FA Service or the FA system according to the instructions located in the *FA Services Configuration to Support IPSec* section of the *Service Configurations* chapter in this guide.

Step 6 Configure one or more transform sets for the HA system according to the instructions located in the *Transform Set Configuration* chapter of this guide.

The transform set(s) must be configured in the same context as the HA service.

Step 7 Configure one or more ISAKMP policies or the HA system according to the instructions located in the *ISAKMP Policy Configuration* chapter of this guide.

The ISAKMP policy(ies) must be configured in the same context as the HA service.

Step 8 Configure an ipsec-isakmp crypto map or the HA system according to the instructions located in the *Dynamic Crypto Map Configuration* section of the *Crypto Maps* chapter of this guide.

The crypto map(s) must be configured in the same context as the HA service.

Step 9 Optional. Configure DPD for the HA to help prevent IPSec tunnel state mismatches between the FA and HA according to the instructions located in the *Dead Peer Detection (DPD) Configuration* section of the *Redundant IPSec Tunnel Fail-Over* chapter of this guide.

Important Though the use of DPD is optional, it is recommended in order to ensure service availability.

Step 10 Configure the HA Service or the HA system according to the instructions located in the *HA Service Configuration to Support IPSec* section in the *Service Configurations* chapter of this guide.

Step 11 Configure the required attributes for RADIUS-based subscribers according to the information located in the *RADIUS Attributes for IPSec-based Mobile IP Applications* chapter of this guide.

Step 12 Save your configuration to flash memory, an external memory device, and/or a network location using the Exec mode command **save configuration**. For additional information on how to verify and save configuration files, refer to the *System Administration Guide* and the *Command Line Interface Reference*.

RADIUS Attributes for IPSec-based Mobile IP Applications

StarOS uses attributes stored in a subscriber's RADIUS profile to determine how IPSec should be implemented.

The table below lists the attributes that must be configured in the subscriber's RADIUS attributes to support IPSec for Mobile IP. These attributes are contained in the following dictionaries:

- 3GPP2
- 3GPP2-835
- Starent
- Starent-835
- Starent-VSA1
- Starent-VSA1-835

Table 3: Attributes Used for Mobile IP IPSec Support

Attribute	Description	Variable
3GPP2-Security-Level	Indicates the type of security that the home network mandates on the visited network.	Integer value: **3** – Enables IPSec for tunnels and registration messages **4** – Disables IPSec
3GPP2 -KeyId	Contains the opaque IKE Key Identifier for the FA/HA shared IKE secret.	Supported value for the first eight bytes is the network-order FA IP address in hexadecimal characters. Supported value for the next eight bytes is the network-order HA IP address in hexadecimal characters. Supported value for the final four bytes is a timestamp in network order, indicating when the key was created, and is the number of seconds since January 1, 1970, UTC.
3GPP2-IKE-Secret	Contains the FA/HA shared secret for the IKE protocol. This attribute is salt-encrypted.	A binary string of 1 to 127 bytes.
3GPP2-S	Contains the "S" secret parameter used to make the IKE pre-shared secret.	A binary string of the value of "S" consisting of 1 to 127 characters
3GPP2- S-Lifetime	Contains the lifetime of the "S" secret parameter used to make the IKE pre-shared secret.	An integer in network order, indicating the time in seconds since January 1, 1970 00:00 UTC Note that this is equivalent to the Unix operating system expression of time.

Implementing IPSec for L2TP Applications

This section provides information on the following topics:

- How IPSec is Used for Attribute-based L2TP Configurations, on page 10
- Configuring Support for L2TP Attribute-based Tunneling with IPSec, on page 11
- How IPSec is Used for PDSN Compulsory L2TP Configuration, on page 11
- Configuring Support for L2TP PDSN Compulsory Tunneling with IPSec, on page 12
- How IPSec is Used for L2TP Configurations on the GGSN, on page 13
- Configuring GGSN Support for L2TP Tunneling with IPSec, on page 14

How IPSec is Used for Attribute-based L2TP Configurations

The following figure and the text that follows describe how IPSec-encrypted attribute-based L2TP sessions are processed by the system.

Table 4: Attribute-based L2TP, IPSec-Encrypted Session Processing 1

Step	
1	A subscriber session arrives at the system.
2	The system attempts to authenticate the subscriber with the AAA server.
3	The profile attributes returned upon successful authentication by the AAA server indicate that session data is to be tunneled using L2TP. In addition, attributes specifying a crypto map name and ISAKMP secret are also supplied indicating that IP security is also required.
4	The system determines that the crypto map name supplied matches a configured crypto map.
5	From the crypto map, the system determines the following: • The map type, in this case dynamic • Whether perfect forward secrecy (PFS) should be enabled for the IPSec SA and if so, what group should be used • IPSec SA lifetime parameters • The name of one or more configured transform set defining the IPSec SA
6	To initiate the IKE SA negotiation, the system performs a Diffie-Hellman exchange of the ISAKMP secret specified in the profile attribute with the specified peer LNS (L2TP Network Server) or security gateway.
7	The system and the LNS or security gateway negotiate an ISAKMP (IKE) policy to use to protect further communications.
8	Once the IKE SA has been negotiated, the system negotiates an IPSec SA with the LNS or security gateway using the transform method specified in the transform sets.

Step	
9	Once the IPSec SA has been negotiated, the system protects the L2TP encapsulated data according to the IPSec SAs established during step 9 and sends it over the IPSec tunnel.

Configuring Support for L2TP Attribute-based Tunneling with IPSec

This section provides a list of the steps required to configure IPSec functionality on the system in support of attributebasedL2TP tunneling. Each step listed refers to a different section containing the specific instructions for completing the required procedure.



Important These instructions assume that the system was previously configured to support subscriber data sessions and L2TP tunneling either as a PDSN or an HA. In addition, with the exception of subscriber attributes, all other parameters configured using this procedure must be configured in the same destination context on the system as the LAC service.

Step 1 Configure one or more transform sets according to the instructions located in the *Transform Set Configuration* chapter of this guide.

Step 2 Configure one or more ISAKMP policies according to the instructions located in the *ISAKMP Policy Configuration* chapter of this guide.

Step 3 Configure an ipsec-isakmp crypto map according to the instructions located in the *Dynamic Crypto Map Configuration* section of the *Crypto Maps* chapter of this guide.

Step 4 Configure the subscriber profile attributes according to the instructions located in the Subscriber Attributes for *L2TP Application IPSec Support* section of the *RADIUS Attributes for IPSec-Based Mobile IP* chapter of this guide.

Step 5 Save your configuration to flash memory, an external memory device, and/or a network location using the Exec mode command **save configuration**. For additional information on how to verify and save configuration files, refer to the *System Administration Guide* and the *Command Line Interface Reference*.

How IPSec is Used for PDSN Compulsory L2TP Configuration

The following figure and the text that follows describe how IPSec-encrypted PDSN compulsory L2TP sessions are processed by the system.

Table 5: PDSN Compulsory L2TP, IPSec-Encrypted Session Processing 2

Step	Description
1	A subscriber session arrives at a PDSN service on the system that is configured to perform compulsory tunneling. The system uses the LAC service specified in the PDSN service's configuration.

Step	Description
2	The LAC service dictates the peer LNS (L2TP Network Server) to use and also specifies the following parameters indicating that IP security is also required: • Crypto map name • ISAKMP secret
3	The system determines that the crypto map name supplied matches a configured crypto map.
4	From the crypto map, the system determines the following: • The map type, in this case dynamic • Whether perfect forward secrecy (PFS) should be enabled for the IPSec SA and if so, what group should be used • IPSec SA lifetime parameters • The name of one or more configured transform set defining the IPSec SA
5	To initiate the IKE SA negotiation, the system performs a Diffie-Hellman exchange of the ISAKMP secret specified by the attribute with the specified peer LNS or security gateway.
6	The system and the LNS or security gateway negotiate an ISAKMP policy (IKE SA) to use to protect further communications.
7	Once the IKE SA has been negotiated, the system negotiates an IPSec SA with the LNS or security gateway.
8	Once the IPSec SA has been negotiated, the system protects the L2TP encapsulated data according to the rules specified in the transform set and sends it over the IPSec tunnel.

Configuring Support for L2TP PDSN Compulsory Tunneling with IPSec

This section provides a list of the steps required to configure IPSec functionality on the system in support of PDSN compulsory L2TP tunneling. Each step listed refers to a different section containing the specific instructions for completing the required procedure.

Important These instructions assume that the system was previously configured to support PDSN compulsory tunneling subscriber data sessions. In addition, all parameters configured using this procedure must be configured in the same destination context on the system as the LAC service.

Step 1 Configure one or more transform sets according to the instructions located in the *Transform Set Configuration* chapter of this guide.

Step 2 Configure one or more ISAKMP policies according to the instructions located in the *ISAKMP Policy Configuration* chapter of this guide.

Step 3 Configure an ipsec-isakmp crypto map according to the instructions located in the *Dynamic Crypto Map Configuration* section of the *Crypto Maps* chapter of this guide.

Step 4 Configure the subscriber profile attributes according to the instructions located in the Subscriber Attributes for *L2TP Application IPSec Support* section of the *RADIUS Attributes for IPSec-Based Mobile IP* chapter of this guide.

Step 5 Save your configuration to flash memory, an external memory device, and/or a network location using the Exec mode command **save configuration**. For additional information on how to verify and save configuration files, refer to the *System Administration Guide* and the *Command Line Interface Reference*.

How IPSec is Used for L2TP Configurations on the GGSN

The following figure and the text that follows describe how IPSec-encrypted attribute-based L2TP sessions are processed by the system.

Table 6: GGSN PDP Context Processing with IPSec-Encrypted L

Step	Description
1	A subscriber session/PDP Context Request arrives at the system.
2	The configuration of the APN accessed by the subscriber indicates that session data is to be tunneled using L2TP. In addition, attributes specifying a crypto map name and ISAKMP secret are also supplied indicating that IP security is also required.
3	The system determines that the crypto map name supplied matches a configured crypto map.

Step	Description
4	From the crypto map, the system determines the following: • The map type, in this case dynamic • Whether perfect forward secrecy (PFS) should be enabled for the IPSec SA and if so, what group should be used • IPSec SA lifetime parameters • The name of one or more configured transform set defining the IPSec SA
5	To initiate the IKE SA negotiation, the system performs a Diffie-Hellman exchange of the ISAKMP secret specified in the profile attribute with the specified peer LNS or security gateway.
6	The system and the LNS or security gateway negotiate an ISAKMP (IKE) policy to use to protect further communications.
7	Once the IKE SA has been negotiated, the system negotiates an IPSec SA with the LNS or security gateway using the transform method specified in the transform sets.
8	Once the IPSec SA has been negotiated, the system protects the L2TP encapsulated data according to the IPSec SAs established during step 9 and sends it over the IPSec tunnel.

Configuring GGSN Support for L2TP Tunneling with IPSec

This section provides a list of the steps required to configure the GGSN to encrypt L2TP tunnels using IPSEC. Each step listed refers to a different section containing the specific instructions for completing the required procedure.



Important These instructions assume that the system was previously configured to support subscriber PDP contexts and L2TP tunneling either as a GGSN. In addition, all parameters configured using this procedure must be configured in the same destination context on the system as the LAC service.

Step 1 Configure one or more transform sets according to the instructions located in the *Transform Set Configuration* chapter of this guide.

Step 2 Configure one or more ISAKMP policies according to the instructions located in the *ISAKMP Policy Configuration* chapter of this guide.

Step 3 Configure an ipsec-isakmp crypto map according to the instructions located in the *Dynamic Crypto Map Configuration* section of the *Crypto Maps* chapter of this guide.

Step 4 Configure APN support for encrypting L2TP tunnels using IPSec according to the instructions located in the *APN Template Configuration to Support L2TP* chapter of this guide.

Step 5 Save your configuration to flash memory, an external memory device, and/or a network location using the Exec mode command **save configuration**. For additional information on how to verify and save configuration files, refer to the *System Administration Guide* and the *Command Line Interface Reference*.

IPSec for LTE/SAE Networks

The Cisco MME (Mobility Management Entity), S-GW (Serving Gateway), and P-GW (Packet Data Network Gateway) support IPSec and IKEv2 encryption using IPv4 and IPv6 addressing in LTE/SAE (Long Term Evolution/System Architecture Evolution) networks. IPSec and IKEv2 encryption enables network domain security for all IP packet switched networks, providing confidentiality, integrity, authentication, and anti-replay protection via secure IPSec tunnels.

Encryption Algorithms

IPSec for LTE/SAE supports the following control and data path encryption algorithms:

- AES-CBC-128 (Advanced Encryption Standard-Cipher Block Chaining-128)
- AES-CBC-256 (Advanced Encryption Standard-Cipher Block Chaining-256)
- DES-CBC (Data Encryption Standard-Cipher Block Chaining)
- 3DES-CBC (Triple Data Encryption Standard-Cipher Bock Chaining)

HMAC Functions

IPSec for LTE/SAE supports the following data path HMAC (Hash-based Message Authentication Code) functions:

- AES-XCBC-MAC-96 (Advanced Encryption Standard-X Cipher Block Chaining-Message Authentication Code-96)
- MD5-96 (Message Digest 5-96)
- SHA1-96 (Secure Hash Algorithm 1-96)

IPSec for LTE/SAE supports the following control path HMAC (Hash-based Message Authentication Code) functions:

- AES-XCBC-MAC-96 (Advanced Encryption Standard-X Cipher Block Chaining-Message Authentication Code-96)
- MD5-96 (Message Digest 5-96)
- SHA1-96 (Secure Hash Algorithm 1-96)
- SHA2-256-128 (Secure Hash Algorithm 2-256-128)

- SHA2-384-192 (Secure Hash Algorithm 2-384-192)
- SHA2-512-256 (Secure Hash Algorithm 2-512-256)

Diffie-Hellman Groups

IPSec for LTE/SAE supports the following Diffie-Hellman groups for IKE and Child SAs (Security Associations):

- Diffie-Hellman Group 1: 768-bit MODP (Modular Exponential) Group
- Diffie-Hellman Group 2: 1024-bit MODP Group
- Diffie-Hellman Group 5: 1536-bit MODP Group
- Diffie-Hellman Group 14: 2048-bit MODP Group
- None: No Diffie-Hellman Group (no perfect forward secrecy)

Dynamic Node-to-Node IPSec Tunnels

IPSec for LTE/SAE enables network nodes to initiate an IPSec tunnel with another node for secure signaling and data traffic between the nodes, enabling up to 64K dynamic, service-integrated IPSec tunnels per chassis. Once established, a dynamic node-to-node IPSec tunnel continues to carry all of the signaling and/or bearer traffic between the nodes. Dynamic node-to-node IPSec for LTE/SAE is supported on the S1-MME interface for signaling traffic between the eNodeB and the MME, on the S1-U interface for data traffic between the eNodeB and the S-GW, and on the S5 interface for data traffic between the S-GW and the P-GW.

Dynamic node-to-node IPSec gets configured using dynamic IKEv2 crypto templates, which are used to specify common cryptographic parameters for the IPSec tunnels such as the encryption algorithm, HMAC function, and Diffie-Hellman group. Additional information necessary for creating node-to-node IPSec tunnels such as revocation lists are fetched dynamically from the IPSec tunnel requests.

For configuration instructions for dynamic node-to-node IPSec, see the configuration chapter in the administration guides for the MME, S-GW, and P-GW.

ACL-based Node-to-Node IPSec Tunnels

Node-to-node IPSec for LTE/SAE can also be configured using crypto ACLs (Access Control Lists), which define the matching criteria used for routing subscriber data packets over an IPSec tunnel. ACL-based node-to-node IPSec tunnels are supported on the S1-MME interface for signaling traffic between the eNodeB and the MME, on the S1-U interface for data traffic between the eNodeB and the S-GW, and on the S5 interface for data traffic between the S-GW and the PGW.

Unlike other ACLs that are applied to interfaces, contexts, or to one or more subscribers, crypto ACLs are applied via matching criteria to crypto maps, which define tunnel policies that determine how IPSec is implemented for subscriber data packets. Prior to routing, the system examines the properties of each subscriber data packet. If the packet properties match the criteria specified in the crypto ACL, the system initiates the IPSec policy dictated by the crypto map. ACL-based node-to-node IPSec tunnels are configured using either IKEv2-IPv4 or IKEv2-IPv6 crypto maps for IPv4 or IPv6 addressing.

Up to 150 ACL-based node-to-node IPSec tunnels are supported on the system, each with one SA bundle that includes one Tx and one Rx endpoint. However, to avoid significant performance degradation, dynamic

node-to-node IPSec tunnels are recommended. If ACL-based node-to-node IPSec tunnels are used, a limit of about ten ACL-based node-to-node IPSec tunnels per system is recommended.

For configuration instructions for ACL-based node-to-node IPSec, see the configuration chapter in the administration guides for the MME, S-GW, and P-GW.

For more information on ACLs, see the *System Administration Guide*.

Traffic Selectors

Per RFC 4306, when a packet arrives at an IPSec subsystem and matches a 'protect' selector in its Security Policy Database (SPD), the subsystem must protect the packet via IPSec tunneling. Traffic selectors enable an IPSec subsystem to accomplish this by allowing two endpoints to share information from their SPDs. Traffic selector payloads contain the selection criteria for packets being sent over IPSec security associations (SAs). Traffic selectors can be created on the P-GW, S-GW, and MME for dynamic node-to-node IPSec tunnels during crypto template configuration by specifying a range of peer IPv4 or IPV6 addresses from which to carry traffic over IPSec tunnels.

For example, consider an eNodeB with an IP address of 1.1.1.1 and an S-GW with a service address of 2.2.2.2. The S-GW is registered to listen for IKE requests from the eNodeBs in the network using the following information:

- Local Address: 2.2.2.2
- Peer Address Network: 1.1.0.0 Mask: 255.255.0.0
- Payload ACL (Access Control List): udp host 2.2.2.2 eq 2123 1.1.0.0 0.0.255.255

When an IKE request arrives the S-GW from eNodeB address 1.1.1.1, the IPSec subsystem converts the payload ACL to: udp host 2.2.2.2 eq 2123 host 1.1.1.1, and this payload becomes the traffic selector for the IPSec tunnel being negotiated.

To properly accommodate control traffic between IPSec nodes, each child SA must include at least two traffic selectors: one with a well-known port in the source address, and one with a well-known port in the destination address. Continuing the example above, the final traffic selectors would be:

- Destination port as well-known port: udp host 2.2.2.2 1.1.0.0 0.0.255.255 eq 2123
- Source port as well-known port: udp host 2.2.2.2 eq 2123 1.1.0.0 0.0.255.255

For ACL-based node-to-node IPSec tunnels, the configured crypto ACL becomes the traffic selector with no modification.

If a TSr (Traffic Selector responder) configuration exists in the crypto template, traffic selector negotiation automatically occurs for the TSr in accordance with RFC 5996 (see exception the note below). If no TSr is configured, the gateway simply respects received traffic selectors and responds with the received traffic selectors. In either case, the gateway can send a maximum of four traffic selectors per TSr.

The negotiation process respects a UE request for a smaller range of IP addresses. Packets are then sent to the target server over the negotiated range.

For additional information on TSr configuration, refer to the *Crypto Template IKEv2-Dynamic Payload Parameters* section in the *Crypto Templates* chapter.

Important For Wireless Security Gateway (WSG) remote access service (RAS), incoming traffic selectors are honored and sent back in the response without negotiation. This exception applies to Security Gateways (SecGWs).

Authentication Methods

IPSec for LTE/SAE includes the following authentication methods:

- **PSK (Pre-Shared Key) Authentication.** A pre-shared key is a shared secret that was previously shared between two network nodes. IPSec for LTE/SAE supports PSK such that both IPSec nodes must be configured to use the same shared secret.
- **X.509 Certificate-based Peer Authentication.** IPSec for LTE/SAE supports X.509 certificate-based peer authentication and CA (Certificate Authority) certificate authentication as described below.

X.509 Certificate-based Peer Authentication

X.509 specifies standard formats for public key certificates, certificate revocation lists, attribute certificates, and a certification path validation algorithm. X.509 certificates are configured on each IPSec node so that it can send the certificate as part of its IKE_AUTH_REQ for the remote node to authenticate it. These certificates can be in PEM (Privacy Enhanced Mail) or DER (Distinguished Encoding Rules) format, and can be fetched from a repository via HTTP or FTP.

CA certificate authentication is used to validate the certificate that the local node receives from a remote node during an IKE_AUTH exchange.

A maximum of sixteen certificates and sixteen CA certificates are supported per system. One certificate is supported per service, and a maximum of four CA certificates can be bound to one crypto template.

For configuration instructions for X.509 certificate-based peer authentication, see the configuration chapter in the administration guides for the MME, S-GW, and P-GW.

The figure below shows the message flow during X.509 certificate-based peer authentication. The table that follows the figure describes each step in the message flow.

For additional information refer to the *IPSec Certificates* chapter of this guide.

Figure 2: X.509 Certificate-based Peer Authentication

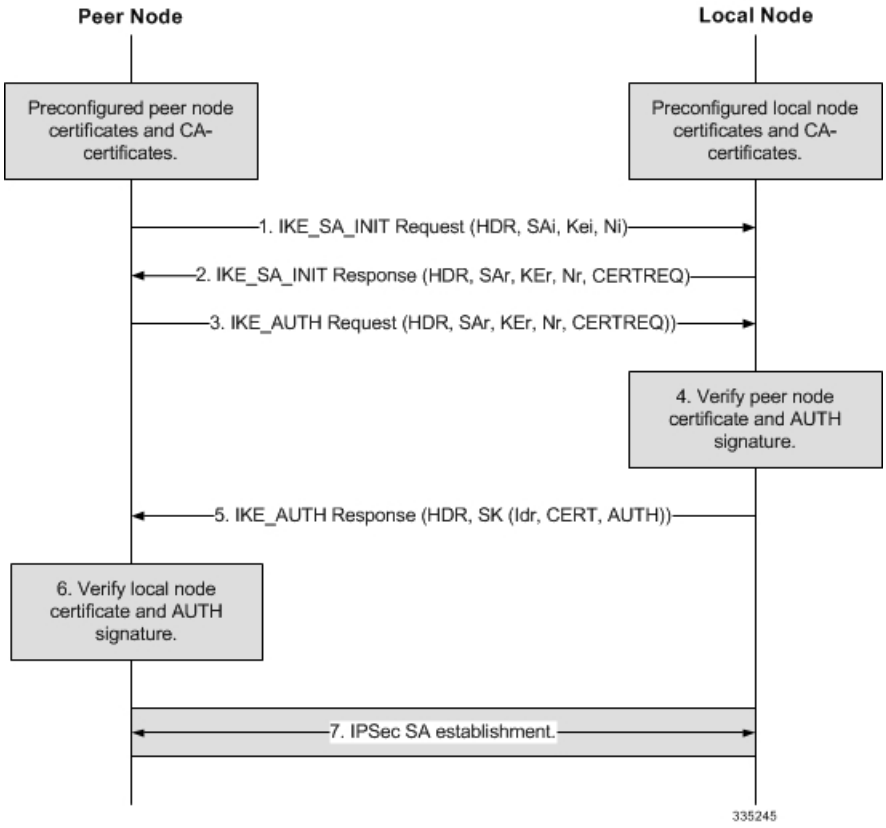


Table 7: X.509 Certificate-based Peer Authentication

Step	Description
1	The peer node initiates an IKEv2 exchange with the local node, known as the IKE_SA_INIT exchange, by issuing an IKE_SA_INIT Request to negotiate cryptographic algorithms, exchange nonces, and perform a Diffie-Hellman exchange with the local node.

Step	Description
2	The local node responds with an IKE_SA_INIT Response by choosing a cryptographic suite from the initiator's offered choices, completing the Diffie-Hellman and nonce exchanges with the peer node. In addition, the local node includes the list of CA certificates that it will accept in its CERTREQ payload. For successful peer authentication, the CERTREQ payload must contain at least one CA certificate that is in the trust chain of the peer certificate. At this point in the negotiation, the IKE_SA_INIT exchange is complete and all but the headers of all the messages that follow are encrypted and integrity-protected.
3	The peer node initiates an IKE_AUTH exchange with the local node by including the IDi payload, setting the CERT payload to the peer certificate, and including the AUTH payload containing the signature of the previous IKE_SA_INIT Request message (in step 1) generated using the private key of the peer certificate. The authentication algorithm used to generate the AUTH payload is also included in the AUTH payload. The peer node also includes the CERTREQ payload containing the list of SHA-1 hash algorithms for local node authentication. For successful server authentication, the CERTREQ payload must contain at least one CA certificate that is in the trust chain of the peer certificate.
4	Using the CA certificate corresponding to the peer certificate, the local node first verifies that the peer certificate in the CERT payload has not been modified and the identity included in the IDi corresponds to the identity in the peer certificate. If the verification is successful, using the public key of the peer certificate, the local node generates the expected AUTH payload and compares it with the received AUTH payload. If they match, the authentication of the peer node is successful. Otherwise, the local node sends an IKEv2 Notification message indicating authentication failure.
5	The local node responds with the IKE_AUTH Response, including the IDr payload, setting the CERT payload to the local node certificate, and including the AUTH payload containing the signature of the IKE_SA_INIT Response message (in step 2) generated using the private key of the local node certificate. The authentication algorithm used to generate the AUTH payload is also included in the AUTH payload.

Step	Description
6	Using the CA certificate corresponding to the local node certificate, the peer node first verifies that the local node certificate in the CERT payload has not been modified. If the verification is successful, using the public key of the local node certificate, the peer generates the expected AUTH payload and compares it with the received AUTH payload. If they match, the local node authentication is successful. This completes the IKE_AUTH exchange.
7	An IPSec SA gets established between the peer node and the local node. If more IPSec SAs are needed, either the peer or local node can initiate the creation of additional Child SAs using a CREATE_CHILD_SA exchange.

Certificate Revocation Lists

Certificate revocation lists track certificates that have been revoked by the CA (Certificate Authority) and are no longer valid. Per RFC 3280, during certificate validation, IPSec for LTE/SAE checks the certificate revocation list to verify that the certificate the local node receives from the remote node has not expired and hence is still valid.

During configuration via the system CLI, one certificate revocation list is bound to each crypto template and can be fetched from its repository via HTTP or FTP.

For additional information refer to the *CRL Fetching* section of the *IPSec Certificates* chapter of this guide.

Child SA Rekey Support

Rekeying of an IKEv2 Child Security Association (SA) occurs for an already established Child SA whose lifetime (either time-based or data-based) is about to exceed a maximum limit. The IPSec subsystem initiates rekeying to replace the existing Child SA. During rekeying, two Child SAs exist momentarily (500ms or less) to ensure that transient packets from the original Child SA are processed by the IPSec node and not dropped.

Child SA rekeying is disabled by default, and rekey requests are ignored. This feature gets enabled in the Crypto Configuration Payload Mode of the system's CLI.

For additional information refer to the *IPSec Certificates* chapter of this guide.

IKEv2 Keep-Alive Messages (Dead Peer Detection)

IPSec for LTE/SAE supports IKEv2 keep-alive messages, also known as Dead Peer Detection (DPD), originating from both ends of an IPSec tunnel. Per RFC 3706, DPD is used to simplify the messaging required to verify communication between peers and tunnel availability. You configure DPD on each IPSec node. You can also disable DPD, and the node will not initiate DPD exchanges with other nodes. However, the node always responds to DPD availability checks initiated by another node regardless of its DPD configuration.

For additional information refer to the *Dead Peer Detection (DPD) Configuration* section of the *Redundant IPSec Tunnel Fail-over* chapter of this guide.

E-UTRAN/EPC Logical Network Interfaces Supporting IPSec Tunnels

The figure below shows the logical network interfaces over which secure IPSec tunnels can be created in an EUTRAN/ EPC (Evolved UMTS Terrestrial Radio Access Network/Evolved Packet Core) network. The table that follows the figure provides a description of each logical network interface.

Figure 3: E-UTRAN/EPC Logical Network Interfaces Supporting IPSec Tunnels

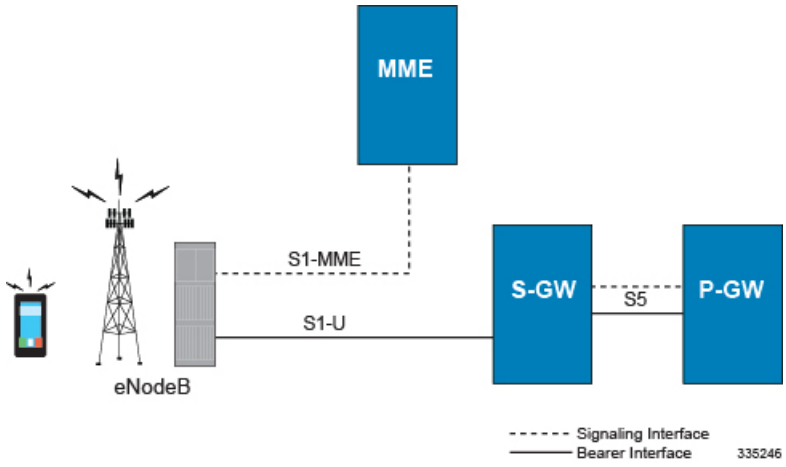


Table 8: E-UTRAN/EPC Logical Network Interfaces Supporting IPSec Tunnels 6

Interface	Description
S1-MME Interface	This interface is the reference point for the control plane protocol between the eNodeB and the MME. The S1-MME interface uses S1-AP (S1- Application Protocol) over SCTP (Stream Control Transmission Protocol) as the transport layer protocol for guaranteed delivery of signaling messages between the MME and the eNodeB (S1). When configured, the S1-AP over SCTP signaling traffic gets carried over an IPSec tunnel. When a subscriber UE initiates a connection with the eNodeB, the eNodeB initiates an IPSec tunnel with the MME, and SCTP signaling for all subsequent subscriber UEs served by this MME gets carried over the same IPSec tunnel. The MME can also initiate an IPSec tunnel with the eNodeB when the following conditions exist: • The first tunnel setup is always triggered by the eNodeB. This is the tunnel over which initial SCTP exchanges occur. • The MME initiates additional tunnels to the eNodeB after an SCTP connection is set up if the MME is multi-homed: a tunnel is initiated from MME's second address to the eNodeB. • The eNodeB is multi-homed: tunnels are initiated from the MME's primary address to each secondary address of the eNodeB. • Both of the prior two conditions: a tunnel is initiated from each of MME's addresses to each address of the eNodeB.
S1-U Interface	This interface is the reference point for bearer channel tunneling between the eNodeB and the S-GW. Typically, the eNodeB initiates an IPSec tunnel with the S-GW over this interface for subscriber data traffic. But the S-GW may also initiate an IPSec tunnel with the eNodeB, if required.

Interface	Description
S5 Interface	This interface is the reference point for tunneling between the S-GW and the P-GW. Based on the requested APN from a subscriber UE, the MME selects both the S-GW and the P-GW that the S-GW connects to. GTP-U data traffic is carried over the IPSec tunnel between the S-GW and P-GW for the current and all subsequent subscriber UEs.

IPSec Tunnel Termination

IPSec tunnel termination occurs during the following scenarios:

- **Idle Tunnel Termination.** When a session manager for a service detects that all subscriber sessions using a given IPSec tunnel have terminated, the IPSec tunnel also gets terminated after a timeout period.
- **Service Termination.** When a service running on a network node is brought down for any reason, all corresponding IPSec tunnels get terminated. This may be caused by the interface for a service going down, a service being stopped manually, or a task handling an IPSec tunnel restarting.
- **Unreachable Peer.** If a network node detects an unreachable peer via Dead Peer Detection (DPD), the IPSec tunnel between the nodes gets terminated. DPD can be enabled per P-GW, S-GW, and MME service via the system CLI during crypto template configuration.
- **E-UTRAN Handover Handling.** Any IPSec tunnel that becomes unusable due to an E-UTRAN network handover gets terminated, while the network node to which the session is handed initiates a new IPSec tunnel for the session.

IPSec for Femto-UMTS Networks



Important HNBGW is not supported from Release 20 and later, and HeNBGW is not supported in Releases 20, 21.0 and 21.1. For more information, contact your Cisco account representative.

The Cisco HNB-GW (Home-NodeB Gateway) supports IPSec and IKEv2 encryption using IPv4 addressing in Femto-UMTS IPSec and IKEv2 encryption enables network domain security for all IP packet-switched networks, providing confidentiality, integrity, authentication, and anti-replay protection via secure IPSec tunnels.

Authentication Methods

IPSec for Femto-UMTS includes the following authentication methods:

- **PSK (Pre-Shared Key) Authentication.** A pre-shared key is a shared secret that was previously shared between two network nodes. IPSec for Femto-UMTS supports PSK such that both IPSec nodes must be configured to use the same shared secret.

- **X.509 Certificate-based Peer Authentication.** IPSec for Femto-UMTS supports X.509 certificate-based peer authentication and CA (Certificate Authority) certificate authentication as described below.

Crypto Map Template Configuration

Use the following example to configure the IPSec profile and crypto template associated with an SeGW and enable IPSec tunneling.

```
configure
   context vpn_ctxt_name
      eap-profile eap_prof_name
         mode authentication-pass-through
         exit
      ip pool ipsec ip_address subnetmask
      ipsec transform-set ipsec_trans_set
      exit
      ikev2 transform-set ikev2_trans_set
      exit
      crypto template crypto_template
         authentication eap-profile eap_prof_name
         exit
         ikev2-ikesa transform set listikev2_trans_set
         payload crypto_payload_name match childsa [ match { ipv4 | ipv6 }
            ip-address-alloc dynamic
            ipsec transform-setlist ipsec_trans_set
            exit
         ikev2-ikesa keepalive-user-activity
         end
configure
   context vpn_ctxt_name
      hnbgw-service hnbgw_svc_name
         security-gateway bind address segw_ip_address crypto-template
crypto_template context segw_ctxt_name
         end
```

Notes:

- *vpn_ctxt_name* is name of the source context in which HNB-GW service is configured
- *segw_ctxt_name* is name of the context in which Se-GW service is configured. By default it takes context where HNB-GW service is configured.
- *hnbgw_svc_name* is name of the HNB-GW service which is to be configured for used for Iuh reference between HNB-GW and HNB

X.509 Certificate-based Peer Authentication

X.509 specifies standard formats for public key certificates, certificate revocation lists, attribute certificates, and a certification path validation algorithm. X.509 certificates are configured on each IPSec node so that it can send the certificate as part of its IKE_AUTH_REQ for the remote node to authenticate it. These certificates

can be in PEM (Privacy Enhanced Mail) or DER (Distinguished Encoding Rules) format, and can be fetched from a repository via HTTP or FTP.

CA certificate authentication is used to validate the certificate that the local node receives from a remote node during an IKE_AUTH exchange.

A maximum of sixteen certificates and sixteen CA certificates are supported per system. One certificate is supported per service, and a maximum of four CA certificates can be bound to one crypto template.

The figure below shows the message flow during X.509 certificate-based peer authentication. The table that follows the figure describes each step in the message flow.

Figure 4: X.509 Certificate-based Peer Authentication

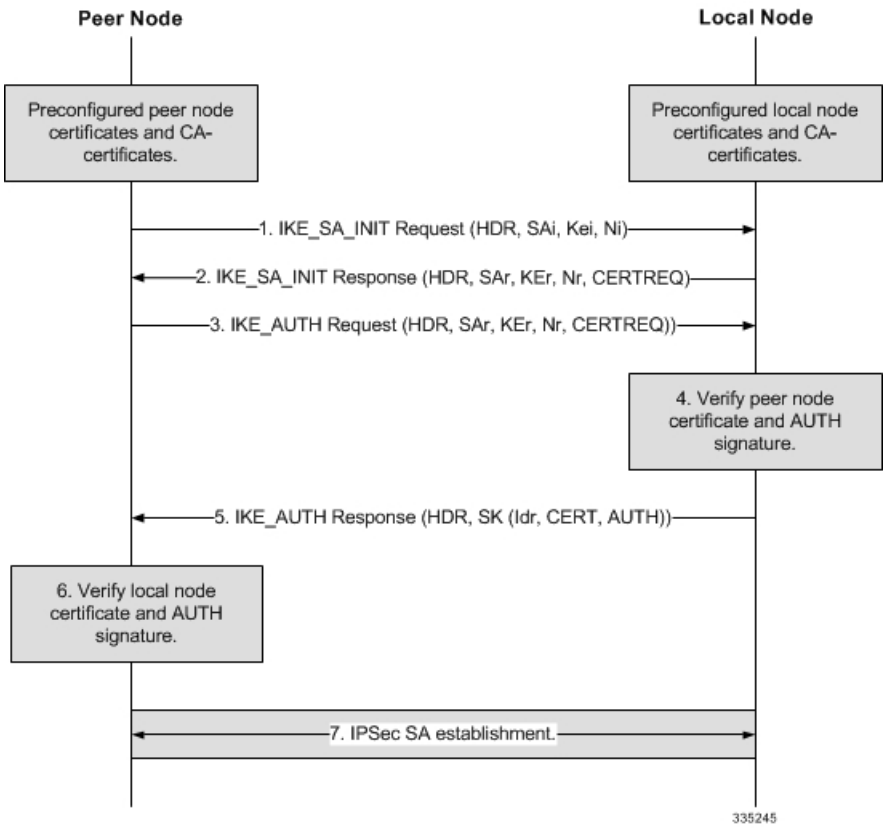


Table 9: X.509 Certificate-based Peer Authentication 9

Step	Description
1	The peer node initiates an IKEv2 exchange with the local node, known as the IKE_SA_INIT exchange, by issuing an IKE_SA_INIT Request to negotiate cryptographic algorithms, exchange nonces, and perform a Diffie-Hellman exchange with the local node.

Step	Description
2	The local node responds with an IKE_SA_INIT Response by choosing a cryptographic suite from the initiator's offered choices, completing the Diffie-Hellman and nonce exchanges with the peer node. In addition, the local node includes the list of CA certificates that it will accept in its CERTREQ payload. For successful peer authentication, the CERTREQ payload must contain at least one CA certificate that is in the trust chain of the peer certificate. At this point in the negotiation, the IKE_SA_INIT exchange is complete and all but the headers of all the messages that follow are encrypted and integrity-protected.
3	The peer node initiates an IKE_AUTH exchange with the local node by including the IDi payload, setting the CERT payload to the peer certificate, and including the AUTH payload containing the signature of the previous IKE_SA_INIT Request message (in step 1) generated using the private key of the peer certificate. The authentication algorithm used to generate the AUTH payload is also included in the AUTH payload. The peer node also includes the CERTREQ payload containing the list of SHA-1 hash algorithms for local node authentication. For successful server authentication, the CERTREQ payload must contain at least one CA certificate that is in the trust chain of the peer certificate.
4	Using the CA certificate corresponding to the peer certificate, the local node first verifies that the peer certificate in the CERT payload has not been modified and the identity included in the IDi corresponds to the identity in the peer certificate. If the verification is successful, using the public key of the peer certificate, the local node generates the expected AUTH payload and compares it with the received AUTH payload. If they match, the authentication of the peer node is successful. Otherwise, the local node sends an IKEv2 Notification message indicating authentication failure.
5	The local node responds with the IKE_AUTH Response, including the IDr payload, setting the CERT payload to the local node certificate, and including the AUTH payload containing the signature of the IKE_SA_INIT Response message (in step 2) generated using the private key of the local node certificate. The authentication algorithm used to generate the AUTH payload is also included in the AUTH payload.

Step	Description
6	Using the CA certificate corresponding to the local node certificate, the peer node first verifies that the local node certificate in the CERT payload has not been modified. If the verification is successful, using the public key of the local node certificate, the peer generates the expected AUTH payload and compares it with the received AUTH payload. If they match, the local node authentication is successful. This completes the IKE_AUTH exchange.
7	An IPSec SA gets established between the peer node and the local node. If more IPSec SAs are needed, either the peer or local node can initiate the creation of additional Child SAs using a CREATE_CHILD_SA exchange.

Certificate Revocation Lists

Certificate revocation lists track certificates that have been revoked by the CA (Certificate Authority) and are no longer valid. Per RFC 3280, during certificate validation, IPSec for LTE/SAE checks the certificate revocation list to verify that the certificate the local node receives from the remote node has not expired and hence is still valid.

During configuration via the system CLI, one certificate revocation list is bound to each crypto template and can be fetched from its repository via HTTP or FTP.

For additional information refer to the *CRL Fetching* section of the *IPSec Certificates* chapter of this guide.

Child SA Rekey Support

Rekeying of an IKEv2 Child Security Association (SA) occurs for an already established Child SA whose lifetime (either time-based or data-based) is about to exceed a maximum limit. The IPSec subsystem initiates rekeying to replace the existing Child SA. During rekeying, two Child SAs exist momentarily (500ms or less) to ensure that transient packets from the original Child SA are processed by the IPSec node and not dropped.

Child SA rekeying is disabled by default, and rekey requests are ignored. This feature gets enabled in the Crypto Configuration Payload Mode of the system's CLI.

For additional information refer to the *IPSec Certificates* chapter of this guide.

IKEv2 Keep-Alive Messages (Dead Peer Detection)

IPSec for LTE/SAE supports IKEv2 keep-alive messages, also known as Dead Peer Detection (DPD), originating from both ends of an IPSec tunnel. Per RFC 3706, DPD is used to simplify the messaging required to verify communication between peers and tunnel availability. You configure DPD on each IPSec node. You can also disable DPD, and the node will not initiate DPD exchanges with other nodes. However, the node always responds to DPD availability checks initiated by another node regardless of its DPD configuration.

For additional information refer to the *Dead Peer Detection (DPD) Configuration* section of the *Redundant IPSec Tunnel Fail-over* chapter of this guide.

IPSec Tunnel Termination

IPSec tunnel termination occurs during the following scenarios:

- **Idle Tunnel Termination.** When a session manager for a service detects that all subscriber sessions using a given IPSec tunnel have terminated, the IPSec tunnel also gets terminated after a timeout period.
- **Service Termination.** When a service running on a network node is brought down for any reason, all corresponding IPSec tunnels get terminated. This may be caused by the interface for a service going down, a service being stopped manually, or a task handling an IPSec tunnel restarting.
- **Unreachable Peer.** If a network node detects an unreachable peer via Dead Peer Detection (DPD), the IPSec tunnel between the nodes gets terminated. DPD can be enabled per P-GW, S-GW, and MME service via the system CLI during crypto template configuration.
- **Network Handover Handling.** Any IPSec tunnel that becomes unusable due to a network handover gets terminated, while the network node to which the session is handed initiates a new IPSec tunnel for the session

x.509 Certificate Configuration

Use the following example to configure the x.509 certificates on the system to provide security certification between FAP and SeGW in Femto-UMTS network.

```
configure
  certificate name x.509_cert_name pem { data pem_data_string | url pem_data_url}
 private-key pem { [encrypted] data PKI_pem_data_string | url PKI_pem_data_url }
     ca-certificate name ca_root_cert_name pem { data pem_data_string | url
pem_data_url }
        exit
     crypto template segw_crypto_template ikev2-dynamic
        authentication local certificate
        authentication remote certificate
        keepalive interval dur timeout dur_timeout
        certificate x.509_cert_name
        ca-certificate list ca-cert-name ca_root_cert_name
        payload crypto_payload_name match childsa [match {ipv4 | ipv6}]
           ip-address-alloc dynamic
           ipsec transform-setlist ipsec_trans_set
           end
configure
  context vpn_ctxt_name
     subscriber default
        ip context-name vpn_ctxt_name
        ip address pool name ip_pool_name
        end
```

Notes:

- *vpn_ctxt_name* is name of the source context in which HNB-GW service is configured.
- *x.509_cert_name* is name of the x.509 certificate where PEM data *pem_data_string* and PKI *PKI_pem_data_string* is configured.

- *ca_root_cert_name* is name of the CA root certificate where PEM data *pem_data_string* is configured for CPE.