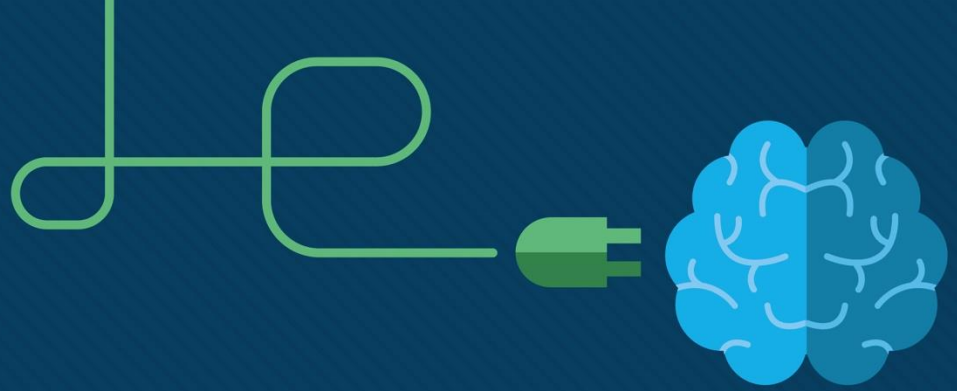

Chapter 16: Overlay Tunnels

Instructor Materials

CCNP Enterprise: Core Networking

Chapter 16 Content

This chapter covers the following content:

Generic Routing Encapsulation (GRE) Tunnels - This section explains GRE and how to configure and verify GRE tunnels.

IPsec Fundamentals - This section explains IPsec fundamentals and how to configure and verify IPsec.

Cisco Location/ID Separation Protocol (LISP) - This section describes the architecture, protocols, and operation of LISP.

Virtual Extensible Local Area Network (VXLAN) - This section describes VXLAN as a data plane protocol that is open to operate with any control plane protocol.

Generic Routing Encapsulation (GRE) Tunnels

- GRE is a tunneling protocol that provides connectivity to a wide variety of network-layer protocols by encapsulating and forwarding packets over an IP-based network.
- GRE can be used to tunnel traffic through a firewall or an ACL or to connect discontiguous networks.
- The most important application of GRE tunnels is that they can be used to create VPNs.

GRE Packet Headers

- When a router encapsulates a packet for a GRE tunnel, it adds new header information (known as encapsulation) to the packet. This new header contains the remote endpoint IP address as the destination.
- The new IP header information enables the packet to be routed between the two tunnel endpoints without inspection of the packet's payload.
- When the packet reaches the remote tunnel endpoint, the GRE headers are removed (known as de-encapsulation) and the original packet is forwarded out of the router.

Figure 16-1 illustrates an IP packet before and after GRE encapsulation. GRE tunnels support IPv4 or IPv6 addresses as an underlay or overlay network.

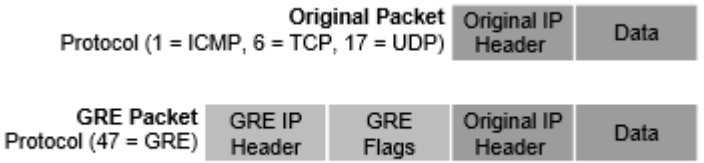

Figure 16-1 *IP Packet Before and After GRE Headers*

GRE Tunnel Configuration

Figure 16-2 illustrates a topology where R1 and R2 are using their respective ISP routers as their default gateways to reach the internet. Example 16-1 shows the routing table on R1.

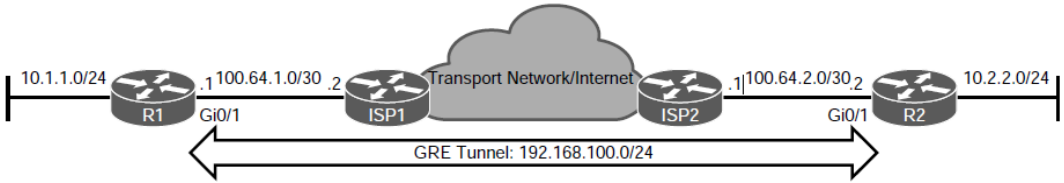

Figure 16-2 *GRE Tunnel Topology*

Example 16-1 *R1's Routing Table Without GRE Tunnel*

```
R1# show ip route
! Output omitted for brevity
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
..
       ia - IS-IS inter area, * - candidate default, U - per-user static route

Gateway of last resort is 100.64.1.2 to network 0.0.0.0

S*    0.0.0.0/0 [1/0] via 100.64.1.2
..
```

GRE Tunnel Configuration (Cont.)

The steps for configuring GRE tunnels are as follows:

Step 1. Create the tunnel interface by using the global configuration command **interface tunnel** *tunnel-number*.

Step 2. Identify the local source of the tunnel by using the interface parameter command **tunnel source** {*ip-address* | *interface-id*}. The tunnel source can be a physical interface or a loopback interface.

Step 3. Identify the remote destination IP address by using the interface parameter command **tunnel destination** *ip-address.*

Step 4. Allocate an IP address to the tunnel interface by using the command **ip address** *ip-address subnet-mask*.

GRE Tunnel Configuration (Cont.)

Optional GRE configuration steps:

Step 5. (Optional) Define the tunnel bandwidth for use by QoS or for routing protocol metrics. Bandwidth is defined with the interface parameter command **bandwidth** [*1-10000000*], which is measured in kilobits per second.

Step 6. (Optional) Specify a GRE tunnel keepalive with the interface parameter command **keepalive** [*seconds* [*retries*]]. The default timer is 10 seconds, with three retries. Tunnel keepalives ensure that bidirectional communication exists between tunnel endpoints to keep the line protocol up.

Step 7. (Optional) Define the IP maximum transmission unit (MTU) for the tunnel interface. Specifying the IP MTU on the tunnel interface has the router perform the fragmentation in advance of the host having to detect and specify the packet MTU. IP MTU is configured with the interface parameter command **ip mtu** *mtu*.

GRE Tunnel Configuration (Cont.)

Example 16-2 provides a GRE tunnel configuration for R1 and R2, following the steps for GRE configuration listed earlier.

With this configuration, R1 and R2 become direct OSPF neighbors over the GRE tunnel and learn each other's routes.

R1

```
interface Tunnel100
 bandwidth 4000
 ip address 192.168.100.1 255.255.255.0
 ip mtu 1400
 keepalive 5 3
 tunnel source GigabitEthernet0/1
 tunnel destination 100.64.2.2
!
router ospf 1
 router-id 1.1.1.1
 network 10.1.1.1 0.0.0.0 area 1
 network 192.168.100.1 0.0.0.0 area 0
!
ip route 0.0.0.0 0.0.0.0 100.64.1.2
```

R2

```
interface Tunnel100
 bandwidth 4000
 ip address 192.168.100.2 255.255.255.0
 ip mtu 1400
 keepalive 5 3
 tunnel source GigabitEthernet0/1
 tunnel destination 100.64.1.1
!
router ospf 1
 router-id 2.2.2.2
 network 10.2.2.0 0.0.0.255 area 2
 network 192.168.100.2 0.0.0.0 area 0

ip route 0.0.0.0 0.0.0.0 100.64.2.1
```

Example 16-2 *Configuring GRE*

GRE Tunnel Verification

The state of the GRE tunnel can be verified with the command **show interface tunnel** *number*. Example 16-3 shows output from this command.

Example 16-3 *Displaying GRE Tunnel Parameters*

```
R1# show interfaces tunnel 100 | include Tunnel.*is|Keepalive|Tunnel s|Tunnel p
Tunnel100 is up, line protocol is up
  Keepalive set (5 sec), retries 3
  Tunnel source 100.64.1.1 (GigabitEthernet0/1), destination 100.64.2.2
  Tunnel protocol/transport GRE/IP
```

GRE Tunnel Verification (Cont.)

Additional commands to verify the status of a GRE tunnel include **show ip route** and **traceroute.** Examples 16-4 and 16-5 show the output of these commands when the GRE tunnel is active.

Example 16-4 *R1 Routing Table with GRE*

```
R1#show ip route
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area

! Output omitted for brevity

Gateway of last resort is 100.64.1.2 to network 0.0.0.0

S*      0.0.0.0/0 [1/0] via 100.64.1.2
        1.0.0.0/32 is subnetted, 1 subnets
C          1.1.1.1 is directly connected, Loopback0
        10.0.0.0/8 is variably subnetted, 3 subnets, 2 masks
C          10.1.1.0/24 is directly connected, GigabitEthernet0/3
L          10.1.1.1/32 is directly connected, GigabitEthernet0/3
O  IA      10.2.2.0/24 [110/26] via 192.168.100.2, 00:17:37, Tunnel100
        100.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
C          100.64.1.0/30 is directly connected, GigabitEthernet0/1
L          100.64.1.1/32 is directly connected, GigabitEthernet0/1
        192.168.100.0/24 is variably subnetted, 2 subnets, 2 masks
C          192.168.100.0/24 is directly connected, Tunnel100
L          192.168.100.1/32 is directly connected, Tunnel100
```

Example 16-5 *Verifying the Tunnel*

```
R1# traceroute 10.2.2.2 source 10.1.1.1
Tracing the route to 10.2.2.2
  1 192.168.100.2 3 msec 5 msec *
```

Problems with Overlay Networks

Recursive routing and outbound interface selection are two common problems with tunnel or overlay networks.

- Recursive routing can occur when the transport network is advertised into the same routing protocol that runs on the overlay network.
- Routers detect recursive route and generate syslog messages.
- Recursive routing problems are remediated by preventing the tunnel endpoint address from being advertised across the tunnel network.

IPsec Fundamentals

- IPsec is a framework of open standards for creating highly secure virtual private networks (VPNs).
- IPsec provides security services such as peer authentication, data confidentiality, data integrity and replay detection.

IPSec Security Services

Table 16-3 *IPsec Security Services*

Security Service	Description	Methods Used
Peer authentication	Verifies the identity of the VPN peer through authentication.	• Pre-Shared Key (PSK) • Digital certificates
Data confidentiality	Protects data from eavesdropping attacks through encryption algorithms. Changes plaintext into encrypted ciphertext.	• Data Encryption Standard (DES) • Triple DES (3DES) • Advanced Encryption Standard (AES) The use of DES and 3DES is not recommended.
Data integrity	Prevents man-in-the-middle (MitM) attacks by ensuring that data has not been tampered with during its transit across an unsecure network.	Hash Message Authentication Code (HMAC): • Message Digest 5 (MD5) algorithm • Secure Hash Algorithm (SHA-1) The use of MD5 is not recommended.
Replay detection	Prevents MitM attacks where an attacker captures VPN traffic and replays it back to a VPN peer with the intention of building an illegitimate VPN tunnel.	Every packet is marked with a unique sequence number. A VPN device keeps track of the sequence number and does not accept a packet with a sequence number it has already processed.

IPSec Packet Headers

IPsec uses two different packet headers to deliver security:

- **Authentication Header** - The authentication header ensures that the original data packet (before encapsulation) has not been modified during transport on the public network. The authentication header does not support encryption, and is not recommended unless authentication is all that is desired.
- **Encapsulating Security Payload (ESP)** - ESP ensures that the original payload (before encapsulation) maintains data confidentiality by encrypting the payload and adding a new set of headers during transport across a public network.

IPSec Packet Transport

Traditional IPsec provides two modes of packet transport:

- **Tunnel mode** - Encrypts the entire original packet and adds a new set of IPsec headers. These new headers are used to route the packet and also provide overlay functions.
- **Transport mode** - Encrypts and authenticates only the packet payload. This mode does not provide overlay functions and routes based on the original IP headers.

Figure 16-3 shows an original packet, an IPsec packet in transport mode, and an IPsec packet in tunnel mode.

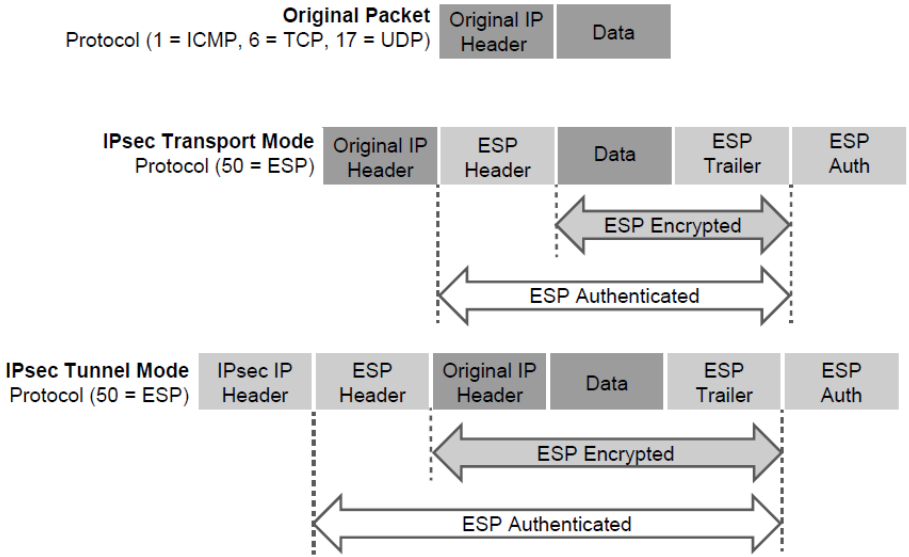

Figure 16-3 *IPsec Transport and Tunnel Encapsulation*

IPSec Encryption, Hashing and Keying

IPsec supports encryption, hashing, and keying methods to provide security services:

- **Data Encryption Standard (DES) -** A 56-bit symmetric data encryption algorithm that can encrypt the data sent over a VPN. This algorithm is very weak and should be avoided.
- **Triple DES (3DES) -** A data encryption algorithm that runs the DES algorithm three times with three different 56-bit keys. Using this algorithm is no longer recommended. The more advanced and more efficient AES should be used instead.
- **Advanced Encryption Standard (AES) -** A symmetric encryption algorithm used for data encryption that was developed to replace DES and 3DES. AES supports key lengths of 128 bits, 192 bits, or 256 bits and is based on the Rijndael algorithm.

IPSec Encryption, Hashing and Keying (Cont.)

- **Message Digest 5 (MD5) -** A one-way, 128-bit hash algorithm used for data authentication. Cisco devices use MD5 HMAC, which provides an additional level of protection against MitM attacks. Using this algorithm is no longer recommended, and SHA should be used instead.
- **Secure Hash Algorithm (SHA) -** A one-way, 160-bit hash algorithm used for data authentication. Cisco devices use the SHA-1 HMAC, which provides additional protection against MitM attacks.
- **Diffie-Hellman (DH) -** An asymmetric key exchange protocol that enables two peers to establish a shared secret key used by encryption algorithms such as AES over an unsecure communications channel.
- **RSA signatures -** A public-key (digital certificates) cryptographic system used to mutually authenticate the peers.
- **Pre-Shared Key -** A security mechanism in which a locally configured key is used as a credential to mutually authenticate the peers

Transform Sets

A transform set is a combination of security protocols and algorithms. During the IPsec SA negotiation, the peers agree to use a particular transform set for protecting a particular data flow.

Transform Type	Transform	Description
Authentication header transform (only one allowed)	ah-md5-hmac	Authentication header with the MD5 authentication algorithm (not recommended)
	ah-sha-hmac	Authentication header with the SHA authentication algorithm
	ah-sha256-hmac	Authentication header with the 256-bit AES authentication algorithm
	ah-sha384-hmac	Authentication header with the 384-bit AES authentication algorithm
	ah-sha512-hmac	Authentication header with the 512-bit AES authentication algorithm

Table 16-4 Allowed Transform Set Combinations

Transform Sets (Cont.)

Transform Type	Transform	Description
ES ESP encryption transform (only one allowed)	esp-aes	ESP with the 128-bit AES encryption algorithm
	esp-gcm esp-gmac	ESP with either a 128-bit (default) or a 256-bit encryption algorithm
	esp-aes 192	ESP with the 192-bit AES encryption algorithm
	esp-aes 256	ESP with the 256-bit AES encryption algorithm
	esp-des esp-3des	ESPs with 56-bit and 168-bit DES encryption (no longer recommended)
	esp-null	Null encryption algorithm
	esp-seal	ESP with the 160-bit SEAL encryption algorithm

Table 16-4 Allowed Transform Set Combinations

Transform Sets (Cont.)

Transform Type	Transform	Description
ESP authentication transform (only one allowed)	esp-md5-hmac	ESP with the MD5 (HMAC variant) authentication algorithm (no longer recommended)
	esp-sha-hmac	ESP with the SHA (HMAC variant) authentication algorithm
IP compression transform	comp-lzs	IP compression with the Lempel-Ziv-Stac (LZS) algorithm

Table 16-4 Allowed Transform Set Combinations

Internet Key Exchange

- Internet Key Exchange (IKE) is a protocol that performs authentication between two end- points to establish security associations (SAs), also known as IKE tunnels.
- There are two versions of IKE: IKEv1 (specified in RFC 2409) and IKEv2 (specified in RFC 7296).
- Internet Security Association Key Management Protocol (ISAKMP) is a framework for authentication and key exchange between two peers to establish, modify, and tear down SAs.
- For Cisco platforms, IKE is analogous to ISAKMP, and the two terms are used interchangeably.

Internet Key Exchange (Cont.)

IKEv1 defines two phases of key negotiation for IKE and IPsec SA establishment:

- **Phase 1 -** Establishes a bidirectional SA between two IKE peers, known as an **ISAKMP SA**. Because the SA is bidirectional, once it is established, either peer may initiate negotiations for phase 2.
- **Phase 2 -** Establishes unidirectional IPsec SAs, leveraging the ISAKMP SA established in phase 1 for the negotiation.

Phase 1 negotiation can occur using **main mode (MM)** or **aggressive mode (AM).** The peer that initiates the SA negotiation process is known as the initiator, and the other peer is known as the responder.

IKE Phase 1 Negotiation Modes

Main mode (MM) consists of six message exchanges and protects information during the negotiation so as not to expose it to eavesdropping.

The six MM message exchanges:

- **MM1 -** First message containing the SA proposals.
- **MM2 -** Sent from the responder with the matching SA proposal.
- **MM3 -** Initiator starts the DH key exchange.
- **MM4 -** Responder sends its own key to the initiator.
- **MM5 -** Initiator starts authentication by sending peer its IP address.
- **MM6 -** Responder sends back a similar packet and authenticates the session. At this point, the ISAKMP SA is established.

IKE Phase 1 Negotiation Modes (Cont.)

Aggressive mode (AM) consists of a three-message exchange and takes less time to negotiate keys between peers. However, it doesn't offer the same level of encryption security provided by MM negotiation, and the identities of the two peers trying to establish a security association are exposed to eavesdropping. These are the three aggressive mode messages:

- **AM1 -** In this message, the initiator sends all the information contained in MM1 through MM3 and MM5.
- **AM2 -** This message sends all the same information contained in MM2, MM4, and MM6.
- **AM3 -** This message sends the authentication that is contained in MM5.

IKE Phase 2 Session Establishment

Phase 2 uses the existing bidirectional IKE SA to securely exchange messages to establish one or more IPsec SAs between the two peers. The method used to establish the IPsec SA is known as **quick mode (QM)**. Quick mode uses a three-message exchange:

- **QM1 -** The initiator (which could be either peer) can start multiple IPsec SAs in a single exchange message. This message includes agreed-upon algorithms for encryption and integrity decided as part of phase 1, as well as what traffic is to be encrypted or secured.
- **QM2 -** This message from the responder has matching IPsec parameters.
- **QM3 -** After this message, there should be two unidirectional IPsec SAs between the two peers.

Perfect Forward Secrecy (PFS) is an additional function for phase 2 that is recommended but is optional because it requires additional DH exchanges that consume additional CPU cycles. The goal of this function is to create greater resistance to crypto attacks and maintain the privacy of the IPsec tunnels by deriving session keys independently of any previous key.

IKEv2

IKEv2 is an evolution of IKEv1 that includes many changes and improvements. In IKEv2, communications consist of request and response pairs called exchanges and are sometimes just called request/response pairs.

1. **IKE_SA_INIT** negotiates cryptographic algorithms, exchanges nonces, and performs a DH exchange. This single exchange is equivalent to IKEv1's first two pairs of messages MM1 to MM4.
2. **IKE_AUTH** authenticates the previous messages and exchanges identities and certificates. Then it establishes an IKE SA and a child SA (the IPsec SA). This is equivalent to IKEv1's MM5 to MM6 as well as QM1 and QM2.

It takes a total of four messages to bring up the bidirectional IKE SA and the unidirectional IPsec SAs, as opposed to six with IKEv1 aggressive mode or nine with main mode.

Differences Between IKEv1 and IKEv2

IKEv1	IKEv2
Exchange Modes	
Main Mode Aggressive Mode Quick Mode	IKE Security Association Initialization (SA_INIT) IKE_Auth CREATE_CHILD_SA
Minimum Number of Messages Needed to Establish IPsec SAs	
Nine with main mode Six with aggressive mode	Four
Supported Authentication Methods	
Pre-Shared Key (PSK) Digital RSA Cert (RSA-SIG) Public Key Both peers must use the same authentication method	Pre-Shared Key (RSA-SIG) Elliptic Curve Digital Signature Cert (ECDSA-SIG) Asymetric authentication is supported. Authentication method can be specified during the IKE_AUTH exchange.

Differences Between IKEv1 and IKEv2 (Cont.)

IKEv1	IKEv2
Next Generation Encryption (NGE)	
Not Supported.	AES-GCM (Galois/Counter Mode) mode SHA-256 SHA-384 SHA-512 HMAC-SHA-256 Elliptic Curve Diffie-Hellman (ECDH) ECDH-384 ECDSA-384
Attack Protection	
MitM protection Eavesdropping protection	MitM protection Eavesdropping protection Anti-DoS protection

Table 16-5 Major Differences Between IKEv1 and IKEv2

IPsec VPN Solutions

Cisco IPsec VPN Solutions:

- **Site-to-Site (LAN-to-LAN) IPsec VPNs -** Site-to-site IPsec VPNs are the most versatile solution for site-to-site encryption because they are the only solution to allow for multivendor interoperability. Difficult to manage in large networks.
- **Cisco Dynamic Multipoint VPN (DMVPN) -** Simplifies configuration for hub-and-spoke and spoke-to-spoke VPNs in Cisco networks. It accomplishes this by combining multipoint GRE (mGRE) tunnels, IPsec, and Next Hop Resolution Protocol (NHRP).
- **Cisco Group Encrypted Transport VPN (GET VPN) -** Developed specifically for enterprises to build any-to-any tunnel-less VPNs (where the original IP header is used) across service provider MPLS networks or private WANs. Provides encryption over private networks which addresses regulatory-compliance guidelines.
- **Cisco FlexVPN -** FlexVPN is Cisco's implementation of the IKEv2 standard, featuring a unified VPN solution that combines site-to-site, remote access, hub-and-spoke topologies and partial meshes (spoke-to-spoke direct). Remains compatible with legacy VPN implementations using crypto maps.
- **Remote VPN Access -** Remote VPN access allows remote users to securely VPN into a corporate network. It is supported on IOS with FlexVPN (IKEv2 only) and on ASA 5500-X and FirePOWER firewalls.

Configuring IPsec VPNs

Even though crypto maps are no longer recommended for tunnels, they are still widely deployed and should be understood. The steps to enable IPsec over GRE using crypto maps are as follows:

- **Step 1.** Configure a crypto ACL to classify VPN traffic by using these commands:

  ip access-list extended *acl _name*

  permit gre host {*tunnel-source IP*} **host** {*tunnel-destination IP*}

- **Step 2.** Configure an ISAKMP policy for IKE SA by using the command **crypto isakmp policy** *priority*. Within the ISAKMP policy configuration mode, encryption, hash, authentication, and the DH group can be specified with the following commands:

  encryption {des | 3des | aes | aes 192 | aes 256}

  hash {sha | sha256 | sha384 | md5}

  authentication {rsa-sig | rsa-encr | pre-share}

  group {1 | 2 | 5 | 14 | 15 | 16 | 19 | 20 | 24}

The keyword **priority** uniquely identifies the IKE policy and assigns a priority to the policy, where 1 is the highest priority.

Configuring IPsec VPNs (Cont.)

- **Step 3.** Configure PSK by using the command **crypto isakmp key** *keystring* **address** *peer-address* [*mask*]. The *keystring* should match on both peers. For *peeraddress* [*mask*], the value 0.0.0.0 0.0.0.0 can be used to allow a match against any peer.
- **Step 4.** Create a transform set and enter transform set configuration mode by using the command **crypto ipsec transform-set** *transform-set-name transform1* [*transform2* [*transform3*]]. In transform set configuration mode, enter the command **mode** [**tunnel | transport**] to specify tunnel or transport modes.
- **Step 5**. Configure a crypto map and enter crypto map configuration mode by using the command **crypto map** *map-name seq-num* [ipsec-isakmp]. In **crypto map configuration mode**, use the following commands to specify the crypto ACL to be matched, the IPsec peer, and the transform sets to be negotiated:

match address *acl-name*

set peer {*hostname* | *ip-address*}

set transform-set *transform-set-name1* [*transform-setname2...transform-set-name6*]

- **Step 6.** Apply a crypto map to the outside interface by using the command **crypto map** *map-name*

Configuring IPsec Site-to-Site VPN

Example 16-7 shows a configuration example for a site-to-site IPsec tunnel using GRE over IPsec with Pre-Shared Key.

Example 16-7 *Configuring GRE over IPsec Site-to-Site Tunnel with Pre-Shared Key*

R1

```
crypto isakmp policy 10
authentication pre-share
hash sha256
encryption aes
group 14
!
crypto isakmp key CISCO123 address 100.64.2.2
!
crypto ipsec transform-set AES_SHA esp-aes esp-sha-hmac
mode transport
!
ip access-list extended GRE_IPSEC_VPN
permit gre host 100.64.1.1 host 100.64.2.2
!
crypto map VPN 10 ipsec-isakmp
match address GRE_IPSEC_VPN
set transform AES_SHA
set peer 100.64.2.2
!
interface GigabitEthernet0/1
 ip address 100.64.1.1 255.255.255.252
crypto map VPN
!
interface Tunnel100
 bandwidth 4000
 ip address 192.168.100.1 255.255.255.0
 ip mtu 1400
 tunnel source GigabitEthernet0/1
 tunnel destination 100.64.2.2

router ospf 1
 router-id 1.1.1.1
 network 10.1.1.1 0.0.0.0 area 1
 network 192.168.100.1 0.0.0.0 area 0
```

R2

```
crypto isakmp policy 10
authentication pre-share
hash sha256
encryption aes
group 14

crypto isakmp key CISCO123 address 100.64.1.1

crypto ipsec transform-set AES_SHA esp-aes esp-sha-hmac
mode transport

crypto ipsec profile IPSEC_PROFILE
 set transform-set AES_SHA

interface GigabitEthernet0/1
 ip address 100.64.2.2 255.255.255.252

interface Tunnel100
 bandwidth 4000
 ip address 192.168.100.2 255.255.255.0
 ip mtu 1400
interface GigabitEthernet0/1
 ip address 100.64.1.1 255.255.255.252
crypto map VPN
!
 tunnel source GigabitEthernet0/1
 tunnel destination 100.64.1.1
 tunnel protection ipsec profile IPSEC_PROFILE

router ospf 1
 router-id 2.2.2.2
 network 10.2.2.0 0.0.0.255 area 2
 network 192.168.100.2 0.0.0.0 area 0
```

Verifying Site-to-Site VPN

Commands that can provide information to verify the operation of a site-to-site VPN include:

- **show interface tunnel100 | include Tunnel protocol**
- **show ip ospf neighbor**
- **show ip route ospf**
- **show crypto isakmp sa**
- **show crypto ipsec sa**

Configuring VTI over IPsec Site-to-Site Tunnel

Example 16-9 shows the configuration changes that need to be made to the GRE over IPsec configuration to enable VTI over IPsec.

The same commands can be used to verify VTI over IPsec as with the IPsec over GRE tunnel.

- **show interface tunnel100 | include Tunnel protocol**
- **show ip ospf neighbor**
- **show ip route ospf**
- **show crypto isakmp sa**
- **show crypto ipsec sa**

Example 16-9 *Configuring VTI over IPsec Site-to-Site Tunnel with Pre-Shared Key*

R1

```
!Remove crypto map from g0/1

interface g0/1
no crypto map VPN

!Configure IPsec transform set

crypto ipsec transform-set AES_SHA esp-aes esp-sha-hmac
mode transport

!Configure IPsec profile

crypto ipsec profile IPSEC_PROFILE
 set transform-set AES_SHA
'
!Enable VTI on tunnel interface and apply IPSec profile
interface Tunnel100
tunnel mode ipsec ipv4
tunnel protection ipsec profile IPSEC_PROFILE
```

R2

```
!Enable VTI on tunnel interface

interface Tunnel100
tunnel mode ipsec ipv4
```

Cisco Location/ID Separation Protocol (LISP)

- The rapid growth of the default-free zone (DFZ), also known as the internet routing table, led to the development of the *Cisco Location/ID Separation Protocol (LISP).*
- LISP is a routing architecture and a data and control plane protocol that was created to address routing scalability problems on the internet.

LISP Architecture Components

Key LISP architecture components:

- **Endpoint identifier (EID) -** An EID is the IP address of an endpoint within a LISP site. EIDs are the same IP addresses in use today on endpoints (IPv4 or IPv6), and they operate in the same way.
- **LISP site -** This is the name of a site where LISP routers and EIDs reside.
- **Ingress tunnel router (ITR) -** ITRs are LISP routers that LISP-encapsulate IP packets coming from EIDs that are destined outside the LISP site.
- **Egress tunnel router (ETR) -** ETRs are LISP routers that de-encapsulate LISP-encapsulated IP packets coming from sites outside the LISP site and destined to EIDs within the LISP site.
- **Tunnel router (xTR) -** xTR refers to routers that perform ITR and ETR functions (which is most routers).
- **Proxy ITR (PITR) -** PITRs are just like ITRs but for non-LISP sites that send traffic to EID destinations.

LISP Architecture Components (Cont.)

- **Proxy ETR (PETR) -** PETRs act just like ETRs but for EIDs that send traffic to destinations at non-LISP sites.
- **Proxy xTR (PxTR) -** PxTR refers to a router that performs PITR and PETR functions.
- **LISP router -** A LISP router is a router that performs the functions of any or all of the following: ITR, ETR, PITR, and/or PETR.
- **Routing locator (RLOC) -** An RLOC is an IPv4 or IPv6 address of an ETR that is internet facing or network core facing.
- **Map server (MS) -** This is a network device (typically a router) that learns EID-to-prefix mapping entries from an ETR and stores them in a local EID-to-RLOC mapping database.
- **Map resolver (MR) -** This is a network device (typically a router) that receives LISP-encapsulated map requests from an ITR and finds the appropriate ETR to answer those requests by consulting the map server.
- **Map server/map resolver (MS/MR) -** When MS and the MR functions are implemented on the same device, the device is referred to as an MS/MR.

LISP Architecture and Protocols

LISP Routing Architecture

LISP separates IP addresses into **endpoint identifiers (EIDs)** and **routing locators (RLOCs).** Unlike in traditional IP routing, endpoints can roam from site to site, and the only thing that changes is their RLOC; the EID remains the same.

LISP Control Plane

The control plane operates in a very similar manner to the Domain Name System (DNS). Just as DNS can resolve a domain name into an IP address, LISP can resolve an EID into an RLOC by sending map requests to the **Map Resolver (MR).**

LISP Architecture and Protocols (Cont.)

LISP Data Plane

Ingress Tunnel Routers (ITRs) LISP-encapsulate IP packets received from EIDs in an outer IP UDP header with source and destination addresses in the RLOC space; in other words, they perform IP-in-IP/UDP encapsulation.

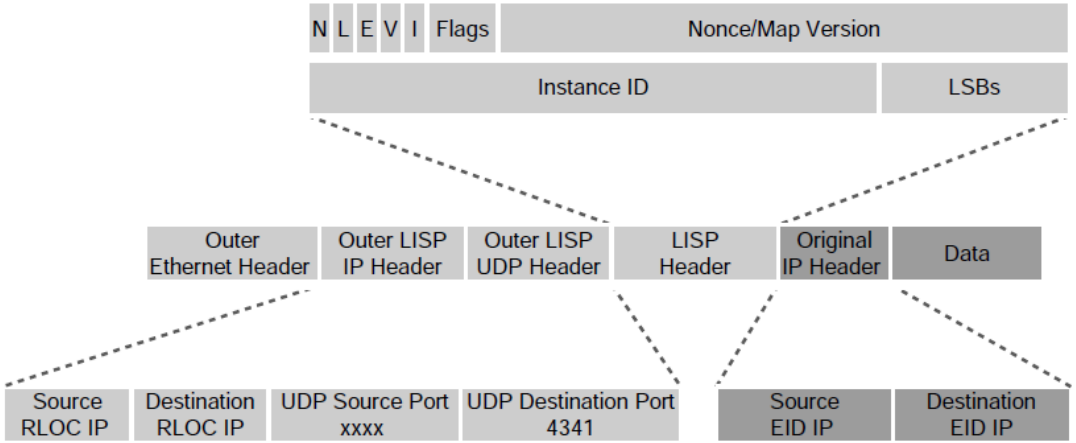

Figure 16-7 *LISP Packet Format*

LISP Map Request and Reply

When an endpoint within a LISP site is trying to communicate to an endpoint outside the LISP site, the ITR needs to perform a series of steps to be able to route the traffic appropriately.

- **Step 1.** The endpoint in LISP Site 1 (host1) sends a DNS request to resolve the IP address of the endpoint in LISP Site 2 (host2.cisco.com). The DNS server replies with the IP address 10.1.2.2, which is the destination EID.
- **Step 2.** The ITR receives the packets from host1 destined to 10.1.2.2. It performs a FIB lookup and evaluates the packet according to the configured forwarding rules.
- **Step 3**. The ITR sends an encapsulated map request to the MR for 10.1.2.2.
- **Step 4.** Because the MR and MS functionality is configured on the same device, the MS mapping database system forwards the map request to the authoritative (source of truth) ETR.
- **Step 5.** The ETR sends to the ITR a map reply message that includes an EID-to-RLOC mapping 10.1.2.2 → 100.64.2.2.
- **Step 6.** The ITR installs the EID-to-RLOC mapping in its local map cache and programs the FIB. It is now ready to forward LISP traffic.

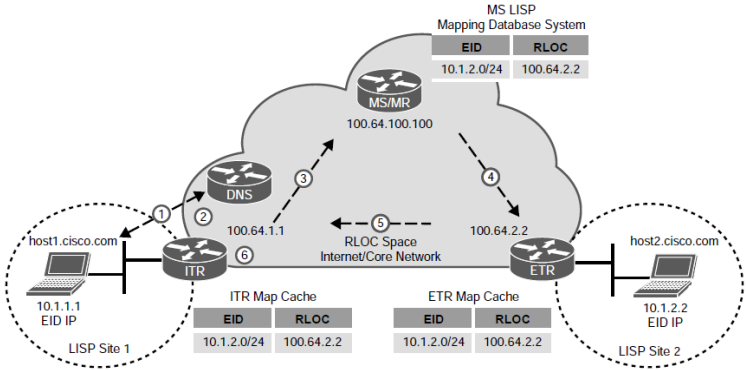

Figure 16-9 *Map Request and Reply*

LISP Data Path

The following steps describe the encapsulation and de-encapsulation process illustrated in Figure 16-10:

- **Step 1.** The ITR receives a packet from EID host1 (10.1.1.1) destined to host2 (10.2.2.2).
- **Step 2.** The ITR performs a FIB lookup and finds a match. It encapsulates the EID packet and adds an outer header with the RLOC IP address from the ITR as the source IP address and the RLOC IP address of the ETR as the destination IP address.
- **Step 3.** ETR receives the encapsulated packet and de-encapsulates it to forward it to host2.

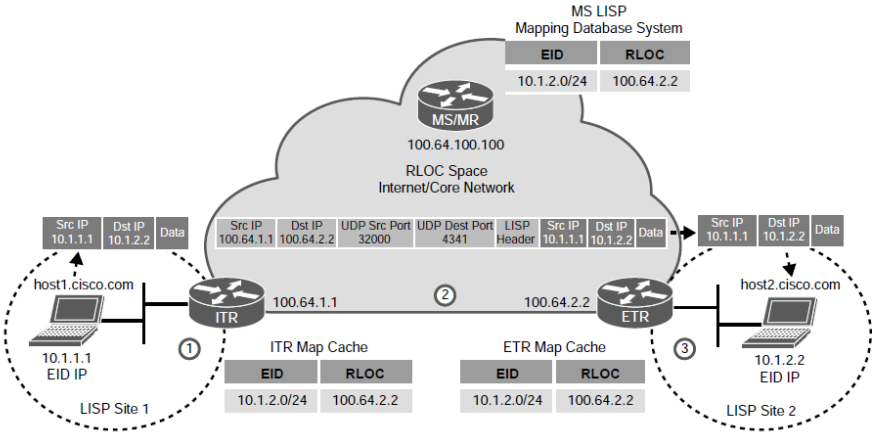

Figure 16-10 *LISP Data Path*

Proxy ETR

The following steps describe the proxy ETR process illustrated in Figure 16-11:

- **Step 1.** host1 perform a DNS lookup for www.cisco.com. It gets a response from the DNS server with IP address 100.64.254.254 and starts forwarding packets to the ITR with the destination IP address.
- **Step 2.** The 100.64.254.254.ITR sends a map request to the MR for 100.64.254.254.
- **Step 3.** The mapping database system responds with a negative map reply that includes a calculated non-LISP prefix for the ITR to add it to its mapping cache and FIB.
- **Step 4.**The ITR can now start sending LISP-encapsulated packets to the PETR.
- **Step 5.** The PETR de-encapsulates the traffic and sends it to www.cisco.com.

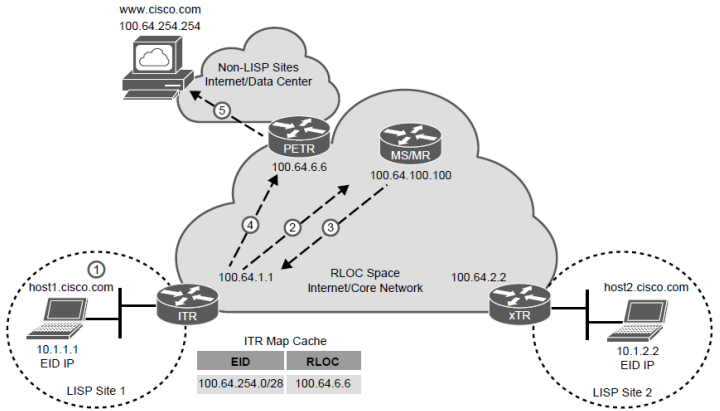

Figure 16-11 *Proxy ETR Process*

Proxy ITR (PITR)

The following steps describe the proxy ITR process illustrated in Figure 16-12:

- **Step 1.** Traffic from www.cisco.com is received by the PITR with the destination IP address 10.1.1.1 from host1.cisco.com.
- **Step 2.** The PITR sends a map request to the MR for 10.1.1.1.
- **Step 3.** The mapping database system forwards the map request to the ETR.
- **Step 4.** The ETR sends a map reply to the PITR with the EID-to-RLOC mapping 10.1.1.1 → 100.64.1.1.
- **Step 5.** The PITR LISP-encapsulates the packets and starts forwarding them to the ETR.
- **Step 6.** The ETR receives the LISP-encapsulated packets, de-encapsulates them, and sends them to host1.

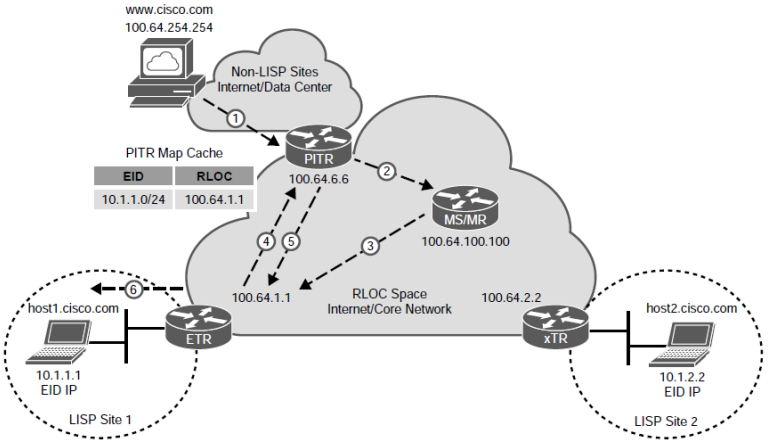

Figure 16-12 *Proxy ITR Process*

Virtual Extensible Local Area Network (VXLAN)

- Server Virtualization has placed an increased demand on legacy network infrastructure.
- Layer 2 networks were not designed to support hundreds of thousands of MAC addresses and tens of thousands of VLANs.
- VXLAN is designed to address the issues being seen in traditional Layer 2 networks.

Issues with Legacy Layer 2 Networks

Virtualization has led to a number of problems with traditional Layer 2 Networks:

- The 12-bit VLAN ID yields 4000 VLANs, which are insufficient for server virtualization.
- Large MAC address tables are needed due to the hundreds of thousands of VMs and containers attached to the network.
- STP blocks links to avoid loops, and this results in a large number of disabled links, which is unacceptable.
- ECMP is not supported.
- Host mobility is difficult to implement.

VXLAN Network Identifier

VXLAN has a 24-bit **VXLAN network identifier (VNI)**, which allows for up to 16 million VXLAN segments (more commonly known as overlay networks) to coexist within the same infrastructure.

- VNI is located in the VXLAN shim header that encapsulates the original inner MAC frame originated by an endpoint. The VNI is used to provide segmentation for Layer 2 and Layer 3 traffic.
- To facilitate the discovery of VNIs over the underlay Layer 3 network, virtual tunnel endpoints (VTEPs) are used.
- Each VTEP has two interfaces:
  - **Local LAN interfaces -** These interfaces on the local LAN segment provide bridging between local hosts.
  - **IP interface -** This is a core-facing network interface for VXLAN. The IP interface's IP address helps identify the VTEP in the network.

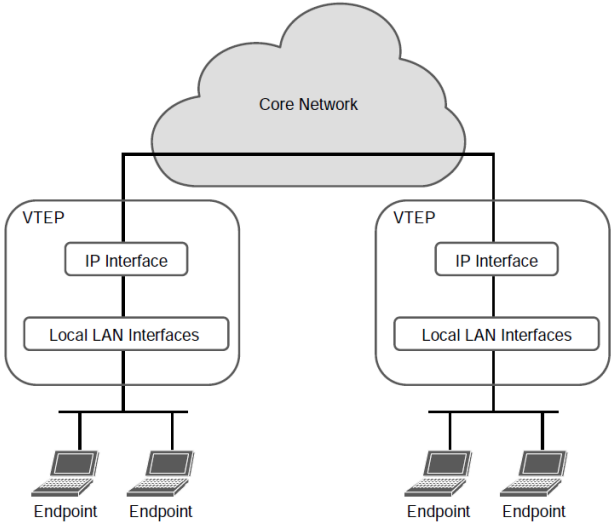

Figure 16-14 *VXLAN VTEP*

VXLAN Headers

There are minor differences between the **Layer 2 LISP** specification and the **VXLAN** specification headers. LISP fields not ported over to VXLAN are reserved for future use.

Cisco Software Defined Access (SD-Access) is an example of an implementation of VXLAN with the LISP control plane.

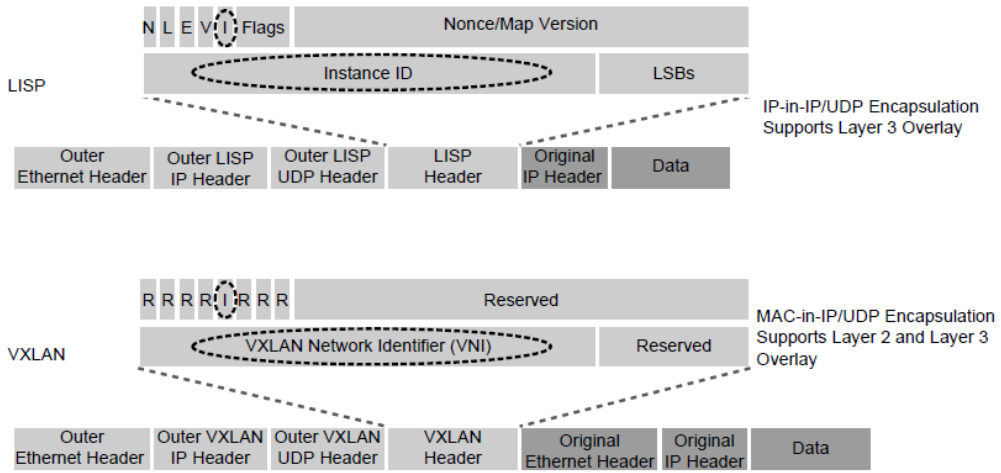

Figure 16-15 *LISP and VXLAN Packet Format Comparison*

Prepare for the Exam

Prepare for the Exam

Key Topics for Chapter 16

Description
Generic Routing Encapsulation (GRE) definition
GRE configuration
IPsec definition
IPsec Security Services
Authentication header
Encapsulating Security Payload (ESP)
IPsec Tunnel and Transport Encapsulation
IPsec security services definitions
Transform sets

Description
Internet Key Exchange (IKE)
IKEv1
IKEv2
Major Differences Between IKEv1 and IKEv2
Cisco IPsec VPN Solutions
Virtual tunnel interface (VTI)
GRE IPsec encryption methods
IPsec over GRE with crypto maps
IPsec over GRE with IPsec profiles

Prepare for the Exam

Key Topics for Chapter 16 (Cont.)

Description
Site-to-Site VTI over IPsec
LISP definition
LISP applications
LISP architecture components
LISP routing architecture
LISP control plane
LISP data plane
LISP map registration and notification
LISP map request and reply

Description
LISP data path
PETR process
PITR process
VXLAN definition
VNI definition
VTEP definition
VXLAN control plane
LISP and VXLAN packet format comparison

Prepare for the Exam

Key Terms for Chapter 16

Term	Term
Egress tunnel router (ETR)	Endpoint identifier (EID)
Ingress tunnel router (ITR)	Internet Key Exchange (IKE)
Internet Protocol Security (IPsec)	Internet Security Association Key Management Protocol (ISAKMP)
LISP Router	LISP site
Map resolver (MR)	Map server (MS)
Map server/map resolver (MS/MR)	Nonce
Overlay network	Proxy ETR (PETR)

Key Terms for Chapter 16 (Cont.)

Term	Term
Proxy ITR (PITR)	Proxy xTR (PxTR)
Routing locator (RLOC)	Segment
Segmentation	Tunnel router (xTR)
Underlay network	Virtual private network (VPN)
Virtual tunnel endpoint (VTEP)	VXLAN network identifier (VNI)

Command Reference for Chapter 16

Task	Command Syntax
Create a GRE tunnel interface	**interface tunnel** *tunnel-number*
Enable keepalives on a GRE tunnel interface	**keepalive** [*seconds* [*retries*]]
Create an ISAKMP policy	**crypto isakmp policy** *priority*
Create an IPsec transform set	**crypto ipsec transform-set** *transform-set-name transform1* [*transform2* [*transform3*]]
Create a crypto map for IPsec	**crypto map** *map-name seq-num* [**ipsec-isakmp**]

Command Reference for Chapter 16 (Cont.)

Task	Command Syntax
Apply a crypto map to an outside interface	**crypto map** *map-name*
Create an IPsec profile for tunnel interfaces	**crypto ipsec profile** *ipsec-profile-name*
Apply an IPsec profile to a tunnel interface	**tunnel protection** *ipsec profile profile-name*
Turn a GRE tunnel into a VTI tunnel	**tunnel mode ipsec** {**ipv4** | **ipv6**}
Turn a VTI tunnel into a GRE tunnel	**tunnel mode gre** {**ip** | **ipv6**}
Display information about ISAKMP SAs	**show crypto isakmp sa**
Display detailed information about IPsec SAs	**show crypto ipsec sa**


CISCO