CYBERTECH PUBLISHING

ITB15200

701 E. Chocolate Avenue, Suite 200, Hershey PA 17033-1240, USA
Tel: 717/533-8845; Fax 717/533-8661; URL-http://www.igi-pub.com

This paper appears in the publication, **Cryptography and Security Services: Mechanisms and Applications**
authored by **M. Mogollon © 2008, IGI Global**

Chapter XI

VPNS and IPSEC

VPNS and IPSEC

Virtual private networks (VPN) and IPsec are discussed in this chapter. A VPN emulates a private wide area network (WAN) facility using IP networks, such as the public Internet or private IP backbones.

When VPNs are used, the Internet offers the appearance, functionality, and usefulness of a dedicated private network. One of the problems in using the Internet as a WAN is that the Internet is a public network and has relatively little security.

IPsec provides the following security services to VPNs: data origin authentication, access control, confidentiality (encryption), connectionless integrity, rejection of replayed packets (a form of partial sequence integrity), and limited traffic flow confidentiality.

Objectives

- Understand VPN concepts and advantages
- Learn how IPsec provides security services to IP Networks
- Become familiar with IPsec concepts of security associations, security protocols, and key management

Introduction

In RFC 2764 (Gleeson, Lin Heinanen, Armitage, & Malis, 2000), an IP based virtual private network is defined as an “emulation of a private wide area network (WAN) facility using IP facilities, including the public Internet, or private IP backbones.” VPNs are used as the basic transport for connecting corporate data centers, remote offices, mobile employees, telecommuters, customers, suppliers, and business partners. The public network is used as a wide area communications network, and it offers the appearance, functionality, and usefulness of a dedicated private network.

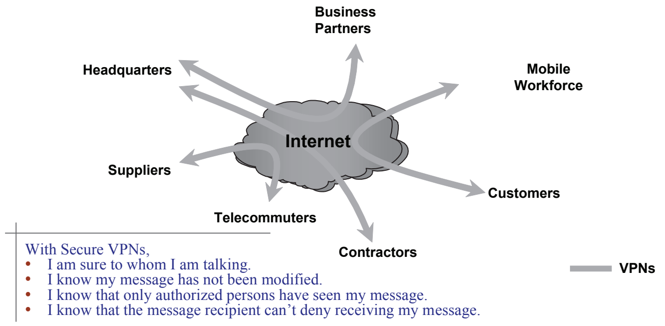
Service providers sold T1 services to corporate clients as a way for the clients to create their own private networks for data traffic. Other technologies such as Frame Relay (FR) and asynchronous transfer mode (ATM) also allow the connection of different sites. The service provider maintains a “cloud” of frame relay connections, and the links are assigned only when needed. As a result, communication prices have gone down considerably.

A T1 leased line normally has a fixed price, with an additional mileage charge per month per mile. Even though frame relay fees do not include a charge for distance and are considerably less expensive than leased lines, monthly fees are still required for the permanent virtual circuits. T1 Internet connections have a monthly fixed price, so one of the main reasons to use the Internet as a corporate WAN is cost savings. There are other compelling arguments for replacing a private network, for example, scalability, responsiveness, and flexibility. Today, most corporations are using the Internet as their corporate WAN because of cost savings and reduced time to set up a connection.

There is a growing need to integrate more closely with partners, suppliers, and customers; there is also a corresponding need to “virtually” extend a company’s geographic reach to include telecommuters and mobile personnel, remote offices and sites, and major vendors and contractors. Therefore, another reason to use the Internet as the corporate WAN network is the savings in long distance charges resulting from, for example, mobile employees not having to call an 800-number to access corporate modem banks. Instead, telecommuters and the mobile force place local calls to the ISP’s POP to connect to the corporate network.

The following are some of the VPN benefits:

Figure 11-1. A corporate virtual private network over the Internet



With Secure VPNs,

- I am sure to whom I am talking.
- I know my message has not been modified.
- I know that only authorized persons have seen my message.
- I know that the message recipient can’t deny receiving my message.

- Ease of use—facilitates electronic communications making corporations more efficient and productive.
- Lower communications cost
- Significant savings resulting from the elimination of long-haul leased lines, 800 numbers or long distance fees, modem banks, and multiple access connections
- Reduction of long distance phone call expenses with use of Voice over IP
- Savings of up to 65% on monthly circuit costs by moving from an FR and ATM environment to an IP VPN
- Lower teleworker connection costs, by as much as 20%-25% per month, over traditional dial-up & ISDN
- Use of standard protocols, IP and IPsec, which provide needed standardization
- Simplification of maintenance and support—reduces scalability issues and management complexity

In VPNs, “virtual” implies that the network is dynamic, with connections set up according to the organization’s needs.

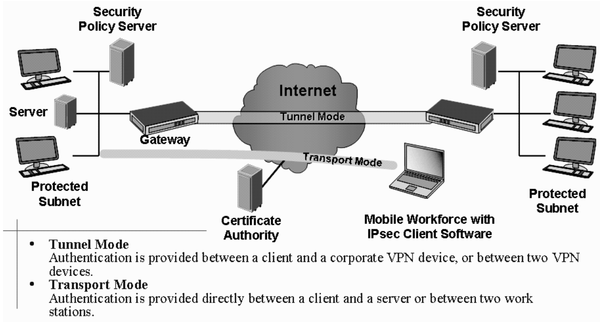
VPN Services

When corporations use the public Internet as a backbone for their communications, there are two alternatives for VPN use: either the service provider provides a secure, managed VPN service or the customer buys the equipment and installs it on his premises. In the first scenario, the service provider provides a service similar to the public switched frame relay or ATM service, and the customer trusts that packets will not be misdirected, modified in transit, or subjected to traffic analysis by unauthorized parties. In the second scenario, the customer does not trust the service provider and implements a VPN using CPE equipment that provides firewall functionality and security. In this case, the service provider is used solely for IP packet transport. In both scenarios, connecting the two VPN endpoints by a virtual tunnel creates security.

A VPN connection is established either by LAN to LAN or client to LAN connections. Gateway switches integrate all of the features needed (firewall, filtering, tunneling, security, bandwidth management and policy management) for high performance, reliable, and secure virtual private networking. Features may include the following:

- Support for point-to-point tunneling protocol (PPTP), L2F, and IPsec with Internet key exchange and X.509 Digital Certificates
- AES, DES, triple DES and RC4 encryption with MD5 and SHA hashing
- Internal or external LDAP, RADIUS, NT Domains, and token card authentication services

Figure 11-2. VPN applications



- **Tunnel Mode**
  Authentication is provided between a client and a corporate VPN device, or between two VPN devices.
- **Transport Mode**
  Authentication is provided directly between a client and a server or between two work stations.

The paths that the encapsulated packets follow in the Internet VPNs are called *tunnels*, not virtual circuits. Part of the encapsulation process performed by a tunnel endpoint includes adding a new address to the packet; this address is the one corresponding to the other endpoint of the tunnel.

IP Tunneling Mechanisms

There are several types of IP tunneling mechanisms and, depending on their form, they can provide some level of intrinsic data security. IP tunneling mechanisms include IP/IP, generic routing encapsulation (GRE) tunnels, layer 2 tunneling protocol (L2TP), IPsec, and multiprotocol label switching (MPLS). Some of these protocols are not often thought of as tunneling protocols, but they are and they do provide some type of protection.

IPsec is considered the best tunneling protocol for IP networks because it provides strong security services such as encryption, authentication, and key management.

L2TP is designed to transport point-to-point protocol (PPP) packets, and thus can be used to carry multiprotocol traffic, since PPP itself is multiprotocol. IP/IP and IPsec tunnels have no such protocol identification field, since the traffic being tunneled is assumed to be IP. L2TP, and PPP is used more in nonIP multiprotocol environments such as NETBEUI, IPX, and AppleTalk.

IPsec

IPsec provides security services at the IP layer by enabling a system to select required security protocols, determine the algorithm(s) to use for the service(s), and put in place any cryptographic keys required to provide the requested services.

IPsec can be used to protect one or more paths between a pair of communicating hosts, between a pair of communicating security gateways, or between a security gateway and a host. The term *security gateway* refers to an intermediate system that implements IPsec protocols. For example, a router or a firewall implementing IPsec is a security gateway.

IPsec provides the following security services: data origin authentication, access control, confidentiality (encryption), connectionless integrity, rejection of replayed packets (a form of partial sequence integrity), and limited traffic flow confidentiality.

IPsec Architecture

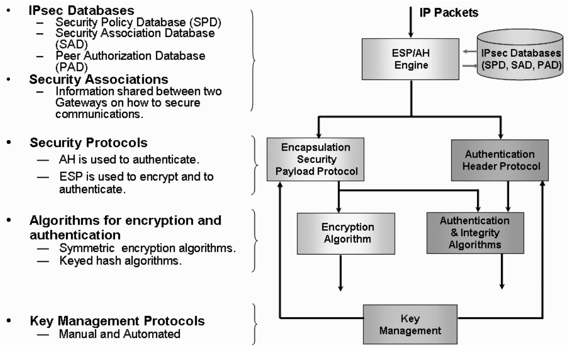
IPsec is a suite of protocols tied together to provide security services. Figure 11-3 shows the different architecture levels. Several documents address details of the IPsec architecture. RFC 4301 (Kent & Seo, 2005), for example, defines the basic architecture for IPsec-compliant systems, the conventions for naming payload formats, exchange types, and security-relevant information such as security policies or cryptographic algorithms and modes. The RFC 4301 also describes the interoperability of all of these elements.

IPsec Databases

There are three nominal databases in IPsec:

- **Security policy database (SPD):** Specifies the initiator and responder IP traffic policies.
- **Security association database (SAD):** Contains the parameters that are associated with each established security association. Each SA has an entry in SAD. The following data is included in SAD: security parameter index (SPI); encapsulated security payload

Figure 11-3. IPsec architecture


• IPsec Databases
– Security Policy Database (SPD)
– Security Association Database (SAD)
– Peer Authorization Database (PAD)
• Security Associations
– Information shared between two Gateways on how to secure communications.
• Security Protocols
— AH is used to authenticate.
— ESP is used to encrypt and to authenticate.
• Algorithms for encryption and authentication
— Symmetric encryption algorithms.
— Keyed hash algorithms.
• Key Management Protocols
— Manual and Automated
IP Packets
ESP/AH Engine
IPsec Databases (SPD, SAD, PAD)
Encapsulation Security Payload Protocol
Authentication Header Protocol
Encryption Algorithm
Authentication & Integrity Algorithms
Key Management

(ESP) encryption and integrity algorithms, key mode, IV, and so forth; authentication header (AH) authentication algorithm, MAC keys, and so forth.; SA lifetime; and IPsec protocol mode, tunnel or transport, applied to the SA.

- **Peer authorization database (PAD):** Provides the link between the SPD and a security management protocol (such as IKE) and the SPD.

Security Protocols

The security protocols consist of the IP Authentication Header (AH), RFC 4302 (Kent, 2005a), which is used to authenticate, and the IP Encapsulated Security Payload (ESP), RFC 4303 (Kent, 2005b), which is used to encrypt and to authenticate.

Security Associations (SA)

Security associations are created when information is shared between two gateways on how to secure a communication. SAs have three parameters:

- Security parameter index (SPI)
- IP destination address
- Security protocol ID, which identifies whether the SA is AH or ESP

Cryptographic Algorithms for Authentication and Encryption

RFC 4305 defines the mandatory, default algorithm for use with AH and ESP. RFC 4307 (Schiller, 2005) defines the mandatory algorithm for use with IKEv2. Each cryptographic algorithm has a separate RFC. AES, triple DES, and other symmetric encryption algorithms are used to encrypt the data. Keyed hash algorithms are used for authentication and integrity.

Key Management Protocols

The key management protocols are described in the Internet Key Exchange (IKEv2), RFC 4306 (Hoffman, 2005).

IPsec Protocols

Figure 11-4 shows the relationship of the IPsec protocols.

The following list defines some of the protocols:

1. Security architecture for IP, RFC 4301 (Kent & Seo, 2005)
2. Security protocols
   a. IP Authentication Header, RFC 4302 (Kent, 2005a)
   b. IP Encapsulating Security Payload (ESP), RFC 4303 (Kent, 2005b)
3. Algorithms for authentication and encryption—a separate RFC for each algorithm.
   a. Internet Key Exchange Version 2 (IKEv2), RFC 4306 (Hoffman, 2005)—A separate RFC for each algorithm

IPsec Negotiation

Refer to Figure 11- 5 for a description of IPsec negotiation.

Outbound Packet

1. The application calls the TCP/IP stack.
2. The TCP/IP packet is captured by the unprotect-protect engine.
3. After checking out the packet in the security policy database, the unprotect-protect engine determines whether it needs to be protected or allowed to bypass IPsec. In general, packets are selected for one of three processing modes based on IP address and transport layer header information matched against entries in the database (SPD). Each packet is either protected using IPsec services, discarded, or allowed to bypass IPsec protection.

Figure 11-4. IPsec protocol

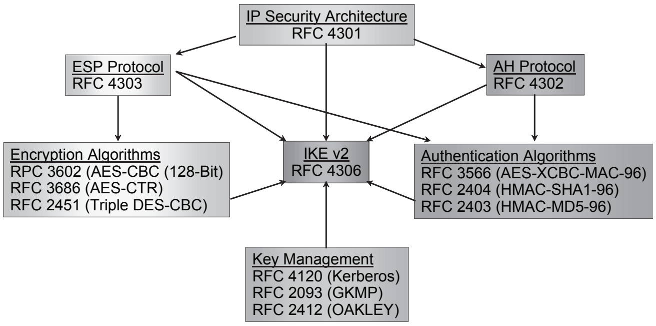
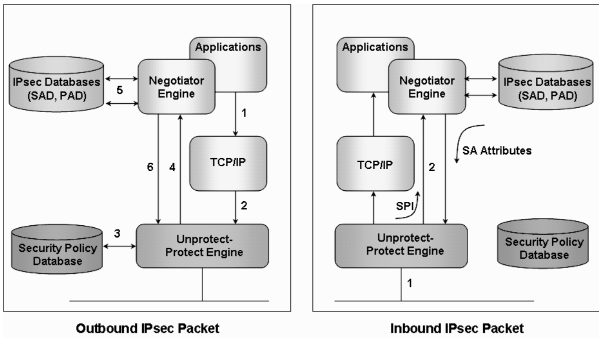


Figure 11-5. IPsec negotiation



4. If a packet needs to be protected, the unprotect-protect engine passes the address on to the negotiator engine that checks the address in the security association (SA) and the security parameter index (SPI).
5. The negotiator engine looks up the SA and SPI in its internal database. If an SA has not been negotiated for that specific address, then the negotiator triggers the creation of an SA by initiating an IKE negotiation with the peer address.
6. Once that negotiation is complete, the SPI and SA are passed on to the unprotect-protect engine. Now the unprotect-protect engine protects all packets sent to that address with keys negotiated by the negotiator.

Inbound Packet

1. If an incoming packet comes in to the port reserved for the IKE negotiation (port 500 or 4500), and no SA has been negotiated with that incoming address, then the unprotect-protect engine will pass all of the IKE packets on to the negotiator.
2. If the arriving packet has a security parameter index (SPI) associated with it, the SA associated with that SPI is retrieved from the IPsec databases. If the SPI is not in the database, then the packet can be rejected.
3. If the arriving packet does not have an SPI embedded in it, the unprotect-protect engine can presume that the packet does not have an SA associated with it. Since there is no SA associated with the packet, it can be rejected.

Security Associations

An SA associates security parameters with the traffic to be protected. It can further be said that an SA describes the security parameters agreed upon between a sender and a receiver, such as a host or gateway, on how to secure a communication.

When a connection is established between a source and its destination, the two need to agree on, among other things, the encryption and authentication algorithms, the crypto keys, the key sizes, key lifetimes, how to exchange keys, the initialization values, and other related security parameters. Once the SA for a specific connection is defined, it is assigned an index, the security parameter index (SPI), and stored in a database, the security policy database (SPD). At the database, the source and destination IP addresses are added to the SA.

The information contained in an SA is grouped into three parameters:

- **Security parameter index (SPI):** The idea of the SPI is to be able to associate an SA with a particular connection, so, if in the future a connection is established between the same source and destination, it is not necessary to agree on a new security association because all the information is stored in the SPD with a binding SPI.
- **IP destination address:** This is the IP address of the end-user or a gateway such as a firewall or a router. In principle, the destination address may be a unicast address, an IP broadcast address, or a multicast group address. However, IPsec SA management mechanisms are defined currently only for unicast SAs.
- S**ecurity protocol ID:** The security protocol ID indicates whether the security protocol is an Encapsulation Security Payload (ESP) or an Authentication Header (AH) protocol.

Security Protocols

IPsec provides mechanisms to provide security services to IP and upper layer protocols (e.g., UDP or TCP). IPsec protects IP datagrams by defining a security protocol in an SA. The SA associated with a connection could be an encapsulating security payload (ESP), or an authentication header (AH), but not both. If both AH and ESP protection are applied to a connection, then two (or more) SAs are created to provide protection to the connection. To secure typical, bidirectional communication between two hosts, or between two security gateways, two security associations (one in each direction) are required. Both ESP and AH security protocols support two modes of operation,i.e., transport or tunnel mode.

The AH protocol, RFC 4302 (Kent, 2005a), provides connectionless integrity, data origin authentication, and an optional anti-replay service. The ESP protocol, RFC 4303 (Kent, 2005b), may provide confidentiality (encryption), and limited traffic flow confidentiality. It also may provide connectionless integrity, data origin authentication, and an anti-replay service. Both AH and ESP are vehicles for access control, based on the distribution of cryptographic keys and the management of traffic flows relative to these security protocols.

Authentication Header

The AH protocol, RFC 4302 (Kent, 2005a), defines the format for IPsec packets that require data origin authentication, connectionless integrity, and anti-replay services only. The AH does not encrypt the data portion of the packet. AH may be applied alone, in combination with ESP, or in a nested fashion through the use of tunnel mode. Figure 11-6 shows the AH format. A description of each of the different fields is given below.

Next Header

The Next Header is an 8-bit field that identifies the type of header that the next payload after the Authentication Header has.

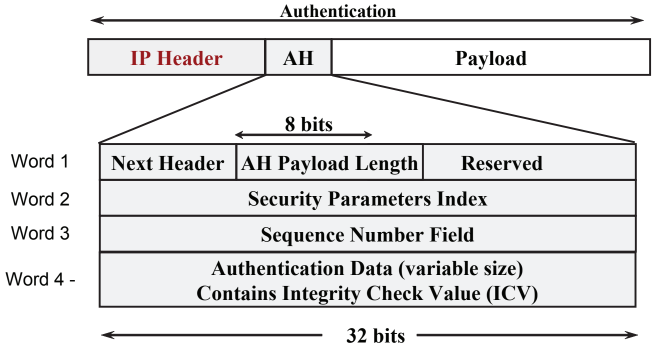
Payload Length

This 8-bit field specifies the length of the AH in 32-bit words (4-byte units), minus 2. In the AH, there are three 32-bit fixed words, so in the standard case of a 96-bit authentication, there are six words in the header, and the Payload field is 4.

Reserved

This 16-bit field is reserved for future use.

Figure 11-6. Authentication header



Security Parameters Index (SPI)

The security parameters index (SPI) tells which security protocols are being used and the algorithms and keys that are included in this field. The SPI is an arbitrary 32-bit value that, in combination with the destination IP address and security protocol (AH), uniquely identifies the security association for the datagram. The SPI value is selected by the destination system at the SA establishment. The Internet Assigned Numbers Authority (IANA), for future use, reserves the values in the range 1 through 255.

Sequence Number

The sequence number tells how many packets have been sent and provides anti-replay protection. This unassigned 32-bit field contains a monotonically increasing counter value (sequence number). It is mandatory and is always present even if the receiver does not elect to enable the anti-replay service for a specific SA. The sender's counter and the receiver's counter are initialized to 0 when an SA is established. The first packet sent using a given SA would have a sequence number of 1. If anti-replay is enabled (the default), the transmitted sequence number must never be allowed to cycle. The sender's counter and the receiver's counter must be reset (by establishing a new SA and, thus, a new key) prior to the transmission of the 2^{32}nd packet in an SA.

Integrity Check Value

This variable-length field contains the integrity check value (ICV) for the packet. The field must be an integral multiple of 32 bits in length and may include explicit padding. This padding is included to ensure that the length of the AH header is an integral multiple of 32 bits (IPv4) or 64 bits (IPv6). All implementations must support such padding.

The authentication algorithm employed for the ICV computation is specified by the SA. For point-to-point communication, suitable authentication algorithms include keyed message authentication codes (HMACs) based on symmetric encryption algorithms (e.g., AES) or on one-way hash functions (e.g., MD5 or SHA1). For multicast communication, one-way hash algorithms combined with asymmetric signature algorithms are appropriate, though performance and space considerations currently preclude use of such algorithms.

The AH ICV is computed over the following:

- IP header fields that are either immutable in transit or that are predictable in value upon arrival at the endpoint for the AH SA
- The AH header including next header, payload length, reserved, SPI, sequence number, and authentication data, and explicit padding bytes, if any
- The upper level protocol data, payload, which is assumed to be immutable in transit

The mandatory-to-implement authentication algorithms are:

- HMAC-SHA-1-96; must be supported
- AES-XCBC-MAC-96; should be supported
- HMAC-MD5-96; may be supported

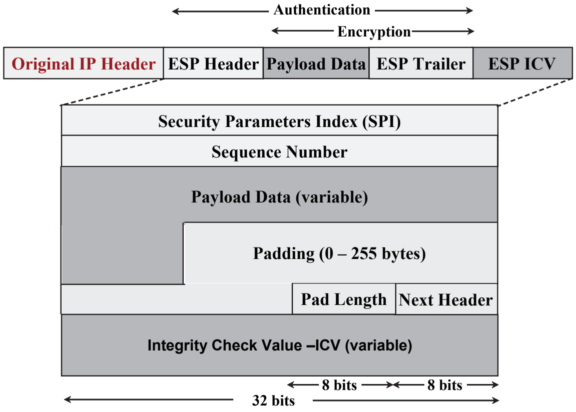
Note: Some HMAC implementation truncates the output H to a given length ***t***, so only part of the hash is outputted. The HMAC notation is as follows: HMAC – Hash algorithm – ***t***. For example, HMAC - SHA1 - 96 is an HMAC that uses SHA1 for its hash function, and the resulting hash is truncated to 96 bits.

Encapsulating Security Protocol (ESP)

RFC 4303 (Kent, 2005b), "ESP protocol," provides the same security services that AH provides (data origin authentication, connectionless integrity, and anti-replay service); it also provides traffic flow confidentiality (encryption). The primary difference between the authentication provided by ESP and the authentication provided by AH is the extent of the coverage. Specifically, ESP does not protect any IP header fields unless those fields are encapsulated by ESP (tunnel mode). The set of services provided by ESP depends on options selected at the time that the security association is established and on the placement of the implementation.

Data origin authentication and connectionless integrity are joint services referred to as *integrity*. Either integrity or confidentiality can be used, but at least one of them must be selected. The anti-replay service may be selected only if integrity service is selected, and its election is solely at the discretion of the receiver. Traffic flow confidentiality requires selection of

Figure 11-7. ESP encapsulation



the tunnel mode and is most effective if implemented at a security gateway, where traffic aggregation may be able to mask true source-destination patterns.

The ESP handles encryption of IP at the packet level using symmetric key encryption. ESP is designed to use any number of encryption algorithms, the most common of which is the Advanced Encryption Standard (AES).

In Figure 11-7, notice the following:

- The ESP header is inserted between the IP header and the rest of the packet.
- The SPI and sequence number field provide the same functions as they do in the AH.
- The TCP portion, data (payload), and ESP trailer are all encrypted.
- ESP provides authentication in the same manner as the AH does.

The following is a description of the different fields:

Security Parameter Index (SPI)

Same as in the AH format.

Sequence Number

Same as in the AH format.

Payload Data

Payload Data is a mandatory, variable-length field containing data described by the Next Header field. ESP is designed for use with symmetric encryption algorithms and because IP packets may arrive out of order, each packet must carry cryptographic synchronization data, for example, an initialization vector, required to allow the receiver to establish cryptographic synchronization for decryption. The cryptographic synchronization data is in the payload.

The mandatory-to-implement encryption algorithms are the following:

- Triple DES-CBC; must be supported
- AES-CBC (128-Bit); should be supported
- AES-CTR; should be supported

Other algorithms, however, such as RC5, IDEA, Three-key Triple IDEA, Cast, and Blowfish, could be used because the Domain of Interpretation (DOI) has assigned identifiers to them.

Padding

Padding is required for the following purposes:

- Some encryption algorithms require the length of the plaintext to be a multiple number of a block size. The padding is used to fill the plaintext to the size required by the algorithm. The plaintext consists of the payload data, the pad length, and the Next Header fields, as well as the padding.
- Padding is also added to the ciphertext to ensure that the resulting number of bits in the ciphertext is a multiple of 32 bits.
- Additional padding may be used to conceal the actual length of the payload, in support of (partial) traffic flow confidentiality.

The sender may add up to 255 bytes of padding.

Pad Length

The Pad Length field indicates the number of pad bytes. The range of valid values is 0-255, where a value of zero indicates that no padding bytes are present. The pad length field is mandatory.

Next Header

The Next Header is an 8-bit field that identifies the type of data contained in the Payload Data field as defined on the Web page of the IANA; for example, a value of 4 indicates IPv4, a value of 41 indicates IPv6. The Next Header field also could indicate an upper layer protocol identifier, for example, a value of 6 indicates TCP.

Integrity Check Value (ICV)

The integrity check value is the same for an ESP format as for an AH format except that for ESP, the ICV is computed over the ESP header, ESP trailer fields, and payload. If the encryption service is selected, the last two fields are in ciphertext form, as the encryption is applied before authentication.

AH and ESP Modes of Operation

IPsec may be implemented in two types of equipment, a host or a security gateway. A security gateway is an intermediate system between two networks; one side of the gateway is viewed as untrusted, the other side as trusted. The gateway implements IPsec on the untrusted interface in order to permit secure communications between hosts on the trusted side and hosts on the untrusted side.

Security services can be provided (1) between a pair of communicating hosts; (2) between a pair of communicating security gateways; or, (3) between a security gateway and a host.

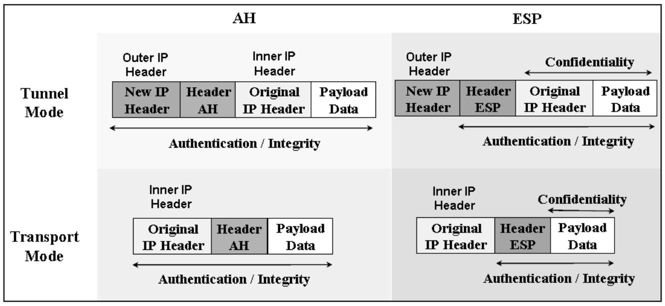
AH and ESP support two modes of operation: the transport mode and the tunnel mode. A transport mode SA is a security association between two hosts. When a security gateway works in transport mode, it acts as a host: the traffic is destined for itself. A tunnel mode SA is a security association between a host and a gateway or between two gateways.

Transport mode is used to protect upper-layer protocols. Tunnel mode is used to protect entire IP packets, meaning that the entire IP packet is encapsulated in another IP packet, and a new IP header is inserted between the outer and inner IP headers.

In a transport mode, the security protocol header appears immediately after the original IP header and before payload data (i.e., any higher layer protocols, e.g., TCP or UDP and Data). In an ESP transport mode, SA provides security services only for the higher layer protocols, not for the original IP header or any extension headers preceding the ESP header. In the case of AH, the protection is extended to the original IP header.

For a tunnel mode SA, an outer header specifies the IPsec end-point and processing destination, plus an inner header that specifies the (apparently) ultimate destination for the packet. The security protocol header appears after the outer IP header, and before the inner IP header. If AH is employed in tunnel mode, portions of the outer IP header are afforded protection (as above), as well as all of the tunneled IP packets, that is, all of the inner IP header is protected, as well as higher layer protocols. If ESP is employed, the protection is afforded only to the tunneled packet, not to the outer header.

Figure 11-8. AH and ESP modes of operation



Algorithms for Encryption and Authentication in IPsec

IPsec allows users to determine which security services to use, the granularity at which a given security protection should be applied, and the encryption and authentication algorithms used. The following is a list of encryption and authentication protocols used in IPsec:

Authentication Header (AH) RFC 4302 (Kent, 2005a):

- HMAC (hash message authentication code)
- SHA1 (RFC 2841)
- MD5 (RFC-1828)

Encapsulation Security Payload (ESP) RFC 4303 (Kent, 2005b):

- Supports only symmetric encryption.
- DES, 3DES, RC5, IDEA, Three-key IDEA, CAST, Blowfish, AES.

Internet Key Exchange RFC 4306 (IKEv2) (Hoffman, 2005), RFC 2412 (Oakley) (Orman, 1998):

- Diffie-Hellman
- Public-key cryptography
- X.509 digital certificates

Internet Key Exchange (IKE v2)

Because IPsec security services use symmetric encryption, it is necessary for both hosts, source and destination, to agree to the mechanisms used to share the secret keys, as well as to the keys that are used for authentication/integrity and encryption services. IPsec supports both manual and automatic distribution of keys. Public key is used for automatic key management, but other automated key distribution techniques may be used.

RFC 4306, IKE v2 (Hoffman, 2005), combines the security concepts of authentication, key management, and security associations to establish the required security for government, commercial, and private communications on the Internet. It does so by defining procedures and packet formats to establish, negotiate, modify, and delete security associations (SA). IKE v2 defines payloads for exchanging key generation and authentication data, thus providing a consistent framework for transferring key and authentication data independent of the key generation technique, encryption algorithm, and authentication mechanism.

A security association (SA) payload indicates a proposal for a set of IPsec encryption algorithms, authentication mechanisms, and key establishment algorithms to be used in IKE, as well as for ESP and/or AH. IKE v2 is not bound to any specific cryptographic algorithm, key generation technique, or security mechanism; the independence from specific security mechanisms and algorithms provides a forward migration path to better mechanisms and algorithms. When improved security mechanisms are developed to counter new attacks against current encryption algorithms, authentication mechanisms and key exchanges would be created; in those situations, IKE v2 would allow the updating of the algorithms and mechanisms without having to develop a completely new IKE or to patch the current one.

A security association normally includes the parameters listed below, but might include additional parameters as well:

- Type of protection used, either ESP or AH
- Authentication algorithm used with AH
- Key(s) used with the authentication algorithm in AH
- Encryption algorithm and mode used with ESP
- Key(s) used with the encryption algorithm in ESP
- Initialization vector for the encryption algorithm used in ESP
- Authentication algorithm and mode used with the ESP transform
- Authentication key(s) used with the authentication algorithm in ESP
- Lifetime of the key used or time when key change should occur
- Hash algorithms to reduce data for signing used
- Information provided about a group over which to do a Diffie-Hellman exchange
- Lifetime of the security association established
- Source address(es) of the security association provided

IKEv2 Algorithm Selection

IKEv1 and IKEv2 provide mechanisms to negotiate which algorithms should be used to ensure interoperability between the initiator and the responder. For IKEv1, the algorithms were listed in RFC 2409, "The Internet Key Exchange (IKEv1)," but for IKE v2, the list was moved from RFC 4306 (Hoffman, 2005), "The Internet Key Exchange (IKEv1)" to RFC 4307 (Schiller, 2005), "IKEv2 Cryptographic Algorithms." If new algorithms are added, RFC 4306 will not need to be changed.

The following features are used by IKE and must be negotiated for the IPsec security association:

- Encryption algorithms to protect data
    - o Must implement 3DES and should implement AES-CBC-128 and AES-CTR-128 modes
- Integrity protection algorithms to produce a fingerprint of the data
    - o Must implement HMAC-SHA1-96, should implement AES-XCBC-96, and may implement HMAC-MD5-96

- Information about which Diffie-Hellman Modular Exponentiation Group (MODP) to use
    - Must implement D-H MODP Group 2 (discrete log 1024 bits), should support D-H Group 14 (2048), and may support D-H elliptic curves over GF [2^{155}] and over GF [2^{185}]
- Pseudorandom function to use
    - Must implement PRF-HMAC-SHA1 (RFC2104), should support PRF-AES-XCBC-PRF-128 (RFC 3664), and may implement PRF-HMAC-MD5 (RFC 2104)

IKE Message Exchanges

In IKE, an initiator proposes one or more cryptographic suites (sets of algorithms) that it is able to support, and the responder mixes-and-matches the suites to create an IKE_SA. The IKE_SA is then used to protect the negotiations for the protocol SA being requested. Two entities (e.g., IPsec servers) can negotiate (and have active) multiple IPsec SAs.

IKE communications consist of pairs of messages, a request followed by a response. The first IKE message exchange always begins with two requests/responses, IKE_SA_INIT (Figure 11-9), steps 1 and 2, and IKE_AUTH, steps 3 and 4. Two entities (e.g., IPsec servers) agree on how to protect further negotiated traffic between them. Subsequent IKE exchanges are CREATE_CHILD or INFORMATIONAL.

In IKE_SA_INIT, the initiator and responder negotiate the use of encryption algorithms by establishing an IKE_SA and, then, by exchanging information for key agreement by sending nonces and Diffie-Hellman values. The agreed keys are used to protect the IKE_AUTH exchange. At this point, the initiator and the responder have agreed on cryptographic key algorithms, but without authenticating each other.

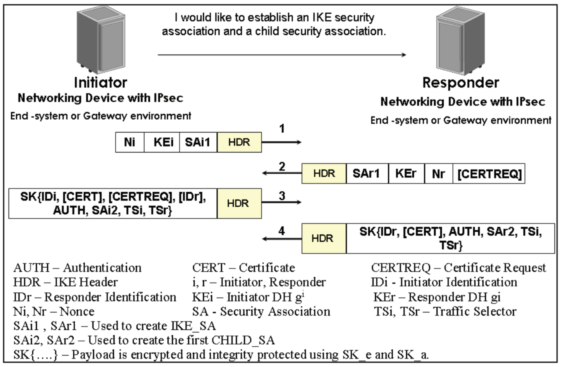
In IKE_AUTH, the initiator and responder authenticate each other using authentication mechanisms such as digital signatures (exchanging certificates), Extensible Authentication Protocol (EAP), or pre-shared keys. In IKE_AUTH, the first IKE_SA and associated IPsec SA, called *child SA*, are created.

The subsequent IKE exchange consists of a single request/response, CREATE_CHILD_SA, that is used to negotiate additional child security associations, which are used to protect additional traffic between the initiator and the responder.

While the IKE_INIT and IKE_AUTH approaches have a higher start-up cost for most simple scenarios, there are several reasons why they would be beneficial for most other cases:

- It takes some time for entities (e.g., IPsec servers) to carry out IKE_INIT and IKE_AUTH, but then time is saved during the subsequent CREATE_CHILD_SAs. This allows multiple child SAs to be established among peers over time without having to start over for each communication.

Figure 11-9. IKE_INIT and IKE_AUTH


I would like to establish an IKE security association and a child security association.
Initiator
Networking Device with IPsec
End -system or Gateway environment
Responder
Networking Device with IPsec
End -system or Gateway environment
1
Ni | KEi | SAi1 | HDR
2
HDR | SAr1 | KEr | Nr | [CERTREQ]
3
SK{IDi, [CERT], [CERTREQ], [IDr], AUTH, SAi2, TSi, TSr} | HDR
4
HDR | SK{IDr, [CERT], AUTH, SAr2, TSi, TSr}
AUTH – Authentication
CERT – Certificate
CERTREQ – Certificate Request
HDR – IKE Header
i, r – Initiator, Responder
IDi - Initiator Identification
IDr – Responder Identification
KEi – Initiator DH g^i
KEr – Responder DH gi
Ni, Nr – Nonce
SA - Security Association
TSi, TSr – Traffic Selector
SAi1 , SAr1 – Used to create IKE_SA
SAi2, SAr2 – Used to create the first CHILD_SA
SK{….} – Payload is encrypted and integrity protected using SK_e and SK_a.

- IKE_INIT establishes an IKE-SA that includes shared secret information that is used to create CHILD_SAs.
- In IKEv2, the first child SA is created on the IKE_AUTH exchange.
- Subsequent CHILD_SAs require only one request/response message in the CREATE_CHILD_SA exchange, making it a very inexpensive operation.

Another IKE exchange is INFORMATIONAL; it is used by either party to convey control messages regarding errors or notifications during the operation of an IKE_SA. Informational exchanges may occur at any time after IKE_AUTH.

IKE_SA_INIT

In **Step 1**, Figure 11-9, the initiator sends an HDR that contains the Security Parameter Index (SPI), the IKE version number, and some message identifiers. These message identifiers include SAi1, which indicates the cryptographic algorithms the initiator supports for the IKE_SA, and the proposed Diffie-Hellman group. The other message identifiers are the KEi payload, which includes the initiator's Diffie-Hellman value, *g^i*, and the initiator's nonce (Ni), which is used to protect against replay attacks. The header (HDR) identifies the initiator's SPI (i.e., the initiator's reference for the IKE_SA to be established), the IKE version number, flags specific to the message, and a message identifier that is used for retransmissions and matching responses to requests. The SAi1 payload includes the supported cryptographic algorithms for the IKE_SA. The SAi1 payload identifies at least one proposal that contains algorithms for encryption, the pseudorandom function, integrity, and the proposed Diffie-Hellman group. The KEi payload includes the initiator's Diffie-Hellman value. The Ni payload contains the initiator's nonce, which is used to protect against replay attacks.

In **Step 2**, Figure 11-9, the responder sends an HDR, which contains the initiator's SPI, the IKE version number, and the same message identifiers used by the initiator. The responder chooses a cryptographic suite by mixing-and-matching the initiator's offered suites and expresses that choice to the initiator in SAr1. The responder also completes the Diffie-Hellman exchange with the KEr, g^r, and sends its nonce in *Nr*. The responder may also (optional) request a specific type of certificate, for example, X.509, by sending the request in [CERTREQ]. No identities are disclosed in the IKE_SA_INIT exchange, other than the IP addresses in the IP headers.

At this point, initiator and responder have negotiated a shared but unauthenticated IKE_SA (SAr1). Also, after the Diffie-Hellman key exchange, each party generates a shared but unauthenticated key, SKEYSEED, from which all keys are derived for that IKE_SA. The keys generated from SKEYSEED are known as the following: SK_e (encryption), and SK_a (message authentication, integrity); SK_d for deriving keys for child SAs; and SK_p for creating AUTH payload in the second request/response exchange. Note that separate SK_e and SK_a keys are generated for each direction. See "Generating Key Material in IKE" section in this chapter.

IKE_SA_AUTH

In **Step 3**, Figure 11-9, the header (HDR) includes the initiator's and the responder's SPI, the IKE version number, and the same message identifiers that were used in the IKE_SA_INIT. The notation SK {…}, also called *encrypted payload*, indicates that the payload is encrypted and integrity protected using SK_e and SK_a.

The following information is included in the initiator's encrypted payload:

- The initiator's identity IDi
- Optional: the initiator's certificate, [Cert], and a list of its trusted root CAs in [CERTREQ]
- The identity of the responder to which the initiator wants to talk by sending [IDr]
- The SAi2 by which the initiator begins negotiation of the first non-IKE security association called *CHILD_SA*, as well as the protocol to be used, for example, Authentication Header (AH) or Encapsulating Security Payload (ESP)
- The traffic selectors TSi and TSr

The whole message is encrypted and its integrity protected using the initiator's SK_e and SK_a. The CHILD_SA is used for ESP and/or AH.

Traffic selectors, TS, allow end points to communicate each other's address and port range, as well as the IP protocol ID that they would like to use. For example, when the initiator sends {192.0.1.0 – 192.0.1.255} as TSi and {192.0.2.0 – 192.0.2.255} as TSr, it means that the initiator would like to tunnel all received information on its IP address range {192.0.1.0 – 192.0.1.255} and would like to tunnel all transmitted information to the responder's IP address range {192.0.2.0 – 192.0.2.255}.

In **Step 4**, Figure 11-9, the header (HDR) includes the initiator's and responder's SPIs, the IKE version number, and the message identifier sent by the initiator in step 3.

The following information is included in the responder's encrypted payload:

- The responder's identity IDr.
- Optional: the responder's certificate, [CERT], if it was requested.
- The AUTH field, which is used by the responder to authenticate itself to the initiator.
- SAr2 to complete the negotiation of CHILD_SA by accepting the proposed algorithms and identifying the negotiated protocol, that is, AH or ESP.
- The traffic selectors with TSi and TSr. If the responder agrees to the traffic selectors proposed by the initiator, the TSi and TSr that the responder sends should be the same as the TSi and TSr sent by the initiator.

All messages exchanged are encrypted and integrity protected with the responder's SK_e and SK_a. The agreed suite of cryptographic algorithms in SAr2 and the shared keys are used to protect the messages in a second message exchange.

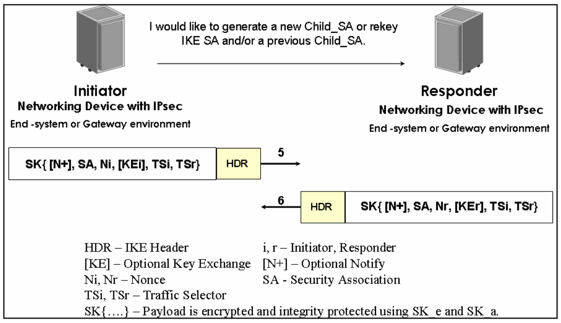
CREATE_CHILD_SAs

The second message exchange consists of a single request/response, which may be initiated by either end, so, in this section, the term *Initiator*, refers to the end point initiating this exchange. The CREATE_CHILD_SA exchange is used to create new CHILD_SAs and to rekey IKE_SAs and CHILD_SAs. All messages are cryptographically protected using the encryption algorithms and keys negotiated in IKE_SA_INIT and IKE_SA_AUTH. However, to enable stronger guarantees of forward secrecy for the key generated for IKE_SA and for CHILD-SA, the CREATE_CHILD_SA request can use additional Diffie-Hellman exchanges to create new keys.

In **Step 5**, Figure 11-10, the initiator sends the header (HDR), which includes the initiator's and the responder's SPIs, the IKE version number, and the message identifiers. The notation SK {…} indicates that the payload is encrypted and its integrity protected using SK_e and SK_a. The initiator (1) sends notify, [N+], which contains additional details for the CHILD_SA (optional step); (2) proposes an SA; (3) sends a nonce in Ni payload; (4) sends a new Diffie-Hellman value, g^i, in Kei payload (optional step); and (5) sends the traffic selectors TSi and TSr. The whole message is encrypted and integrity protected using keys computed from SK_d.

In **Step 6**, Figure 11-10, the responder sends the header (HDR) which includes the initiator's and responder's SPIs, the IKE version number, and the message identifier. The responder (1) sends notify, [N+], which contains additional details for the CHILD_SA (optional step); (2) agrees to the proposed algorithms in an SA payload; (3) sends its nonce in Nr payload; (4) sends a new Diffie-Hellman value, g^r, in Kei payload (optional step); and (5) sends the

Figure 11-10. CREATE CHILD_SA exchange


I would like to generate a new Child_SA or rekey IKE SA and/or a previous Child_SA.
Initiator
Networking Device with IPsec
End -system or Gateway environment
Responder
Networking Device with IPsec
End -system or Gateway environment
SK{ [N+], SA, Ni, [KEi], TSi, TSr}
HDR
5
6
HDR
SK{ [N+], SA, Nr, [KEr], TSi, TSr}
HDR – IKE Header
[KE] – Optional Key Exchange
Ni, Nr – Nonce
TSi, TSr – Traffic Selector
SK{….} – Payload is encrypted and integrity protected using SK_e and SK_a.
i, r – Initiator, Responder
[N+] – Optional Notify
SA - Security Association

traffic selectors TSi and TSr. The whole message is encrypted and integrity protected using keys computed from SK_d.

IKE, ESP and AH security associations use secret keys that should be used only for a limited amount of time and to protect a limited amount of data. When an SA has expired, new security associations can be established by rekeying the IKE_SA or a CHILD_SA.

If CREATE_CHILD_SA is used to rekey IKE_SAs, then the following is exchanged:

Initiator		**Responder**
HDR, SK{ SA, Ni, [Kei]}	→	
	←	HDR, SK{ SA, Nr, [Ker]}

If CREATE_CHILD_SA is used to rekey CHILD_SAs, then the following is exchanged:

Initiator		**Responder**
HDR, SK{ N(REKEY_SA), [N+], SA, Ni, [Kei], TSi, TSr}	→	
	←	HDR, SK{ [N+], SA, Ni, [Kei], TSi, TSr}

Note that the initiator identifies the CHILD_SA being rekeyed in the leading notifying payload, N (REKEY_SA).

As stated in RFC 4718, IKE v2 Clarifications, section 5.2, rekeying the IKE_SA establishes new keys for the IKE_SA and resets the Message ID counters, but it does not authenticate the parties (no AUTH or payload is involved). IKEv2 does not have special provisions for reauthentication, so it is done by creating a new IKE_SA from scratch, using a new IKE_SA_INIT and IKE_SA_AUTH.

Informational Exchange in IKE

An information exchange is used by the initiator and responder to convey control messages regarding errors or notifications during the operation of an IKE_SA. Informational exchanges must only occur when an IKE_SA has been created and is, therefore, cryptographically protected with the negotiated keys. The messages included in informational exchanges are notification (N), delete (D), and configuration payloads (CP). If the informational exchange does not contain a message, it becomes a type of "Are you there?" where one end-point is verifying whether or not the other end-point is alive.

An informational exchange is defined as follows:

Initiator		**Responder**
HDR, SK { [N], [D], [CP], . . . }	→	
	←	HDR, SK{ [N], [D], [CP] . . .}

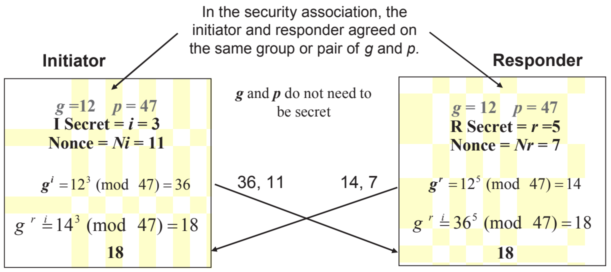
Generating Key Material in IKE

In IKE_SA, four cryptographic algorithms are negotiated: encryption algorithms, integrity protection algorithms, a Diffie-Hellman group, and a pseudorandom function (prf). Key material for all of the cryptographic algorithms used in both IKE_SA and CHILD_SA is always derived as the output of a prf algorithm. In IKEv2, Diffie-Hellman is the only key exchange algorithm used.

The Diffie-Hellman exchange has the following three components: a generator *g*, the modulo *p*, and a secret that in IKEv2 terminology is called *i* or *r*. During IKE_INIT, the initiator and responder exchange Diffie-Hellman information in KEi and KEr. That information includes g^i and g^r, as well as nonces Ni and Nr.

The shared key, SKEYSEED, is calculated by both the initiator and responder from the nonces exchanged and the generated Diffie-Hellman shared secret key, g^{ir}, according to the following formula:

Figure 11-11. Key exchange



Both ends use 11, 7, and 18, as the secret and seed to calculate SKEYSEED

$$SKEYSEED = prf\quad Ni\quad Nr\quad g^{ir}\,)\qquad ,\qquad |\qquad ($$

$$SKEYSEED = prf\,(\,secret, seed\,)$$

$SKEYSEED = prf\ (Ni \,||\, Nr\,, g^{ir}\ )$

Once SKEYSEED is generated by both ends, it is used to calculate seven other keys:

- SK_d for deriving new keys for the CHLD_SA
- SK_ai and SK_ar for integrity protection
- SK_ei and S_Ker for enciphering and deciphering all exchanges
- SK_pi and SK_pr for authentication

SKEYSEED's derivatives are calculated as follows:

{ SK_d || SK_ai || SK_ar || SK_ei || SK_er || SK_pi || SK_pr } = prf+ (SKEYSEED, Ni || Nr || SPIi || SPIr). Note that prf+ is simple the prf of a prf.

The bits for SK_d, SK_ai, SK_ar, SK_ei, SK_er, SK_pi, and SK_pr are taken in order and from the generated bits of prf+.

prf+ (K, S) = T1 || T2 || T3 || T4 || … || Tn

where

- T1 = prf (SKEYSEED, Ni || Nr || SPIi || SPIr || 0x01)
- T2 = prf (SKEYSEED, T1 || Ni || Nr || SPIi || SPIr || 0x02)
- T3 = prf (SKEYSEED, T2 || Ni || Nr || SPIi || SPIr || 0x03)
- T4 = prf (SKESEED, T3 | Ni || Nr || SPIi || SPIr || 0x04)
- Tn = prf (SKESEED, Tn-1 || Ni || Nr || SPIi || SPIr || 0x0n)
- prf (key, seed) is a keyed pseudorandom function, for example, PRF-AES-XCBC-PRF-128 (RFC 3664).
- SPIi and SPIr are the security parameter indexes of the initiator and responder.

Note that each traffic direction uses different keys, so SK_ei and SK_ai are used to protect messages originating from the initiator, and SK_er and SK_ar are used to protect messages originating from the responder. Also, note that because Ni and Nr are used as the keys for the prf, then the nonces should be randomly selected and must be at least half the key size of the negotiated prf.

Generating Key Material for CHILD_SA

If additional CHILD_SAs are created in CREATE_CHILD_SA, the keying material is generated as follows:

KEYMAT = prf+ (SK_d, Ni || Nr), or KEYMAT = prf+ (SK_d, g^{ir} (new) || Ni || Nr)

In the first formula, the seed used in prf+ are the nonces from the CREATE_CHILD_SA exchange, Ni and Nr. In the second formula, the seed also includes a new-shared secret, g^{ir}, from the ephemeral Diffie-Hellman exchange of the CREATE_CHILD_SA exchange.

Integrity and Authentication in IKE

For authentication, IKEv2 uses digital signature algorithms, shared secrets, and EAP methods defined in RFC 3748. See Chapter VII, "Access Authentication," for more information about EAP.

The authentication payload, AUTH, has two types of information, the type of authentication method used and the authentication data. IKEv2 uses the following authentication methods: RSA digital signature, shared key message integrity code, and DSS digital signature. A description of authentication data can be derived by looking at IKE_INIT and IKE_AUTH exchanges.

Initiator		**Responder**
HDR, SAi, Kei, Ni	(Step 1)→	
	←(Step 2)	HDR, SAr, KEr Nr, [CERTREQ]
HDR, SK{ IDi, [CER],		
[CERTREQ], [IDr], AUTH,		
SAi2, TSi, TSr}	(Step 3)→	
		HDR, SK{ IDr, [CERT], AUTH,
	←(Step 4)	SAr2, TSi, TSr}

In **Step 3**, Figure 11-9, the authentication data that the initiator signs in AUTH includes the message sent in step 1, appended to Nr, and the value of prf (SK_pr, IDr'). Note the values of Nr and prf (SK_pr, IDr') are not sent, but included in the message that is signed. IDr' is the responder's ID payload without the header.

In **Step 4**, Figure 11-9, the authentication data that the responder signs in AUTH includes the message sent in step 2, appended to Nr, and the value of prf (SK_pr, IDi'). Note the values of Ni and prf (SK_pr, IDi') are not sent, but included in the message that is signed. IDi' is the initiator's ID payload without the header.

Steps 3 and 4 may include a certificate or certificate authority (CA) that provides evidence of the public-key ownership used to calculate digital signatures. IKEv2 allows an entity initiating communications to indicate which CAs it supports. After selection of a CA, the protocol provides the messages required to support the actual authentication exchange. The protocol provides a facility for identification of different certificate authorities, certificate types (e.g., X.509, PKCS #7, PGP, DNS SIG and KEY records), and the exchange of the certificates identified.

If the initiator would like to use extensible authentication, it does not include the AUTH payload in step 3, meaning that it has not proven its identity, IDi. The responder, then, includes an EAP payload in step 4 and defers sending SAr2, TSi, and TSr until authentication is completed. The initial SA will appear as follows:

Initiator		**Responder**
HDR, SAi, Kei, Ni	→	
	←	HDR, SAr, KEr Nr, [CERTREQ]
HDR, SK{ Idi, [CERTREQ], [IDr], SAi2, TSi, TSr}	→	
	←	HDR, SK{ IDr, [CERT], AUTH, EAP}
HDR, SK{ EAP }	→	
	←	HDR, SK{ EAP (success) }
HDR, SK{ AUTH }	→	
	←	HDR, SK{ AUTH SAr2, TSi, TSr }

Diffie-Hellman Group Descriptors

The ephemeral Diffie-Hellman key exchange is used in IKE to generate keying material, in this way supporting what is called *perfect forward secrecy*. Once a connection is closed, each end point forgets not only the exchanged keys, but also the secrets used in the Diffie-Hellman key calculation.

Three distinct group representations can be used with IKE. The three types are modular exponentiation groups (named MODP), elliptic curve groups over the field GF [2^{n}] (named EC2N), and elliptic curve groups over GF [P] (named ECP). For each representation, many distinct realizations are possible, depending on parameter selection.

RFC 2409, "The Internet Key Exchange (IKEv1)," and RFC 4307 (Schiller, 2005), "IKEv2 Cryptographic Algorithms," specify the following IKE groups:

- **Group 2:** A modular exponentiation group with a 1024-bit modulus
- **Group 14:** A modular exponentiation group with a 2048-bit modulus
- **Group 3:** An elliptic curve group over GF [2^{155}]
- **Group 4:** An elliptic curve group over GF [2^{185}]

RFC 3526 (Kivinen & Kojo, 2003), "More Modular Exponential (MODP) Diffie-Hellman Groups for IKE," specifies stronger Diffie-Hellman groups that are equivalent to AES strength. For example, the 128-bit AES requires a 3200-bit group, and the 192 and 256-bit keys would need groups that are about 8000 and 15400 bits respectively. The following are the new Diffie-Hellman groups proposed in RFC 3526:

- **Group 5:** A modular exponentiation group with a 1536-bit modulus
- **Group 15:** A modular exponentiation group with a 3072-bit modulus
- **Group 16:** A modular exponentiation group with a 4096-bit modulus
- **Group 17:** A modular exponentiation group with a 6144-bit modulus
- **Group 18:** A modular exponentiation group with an 8192-bit modulus

The following are examples of Diffie-Hellman groups:

Group 2: Modular Exponentiation with a 1024 Bit Prime

The generator $g = 2$

$p = 2^{1536} - 2^{1472} - 1 + 2^{64} * \{ [2^{1406} pi] + 741804 \}$.

Its hexadecimal value is:

```
FFFFFFFF FFFFFFFF C90FDAA2 2168C234 C4C6628B 80DC1CD1
29024E08 8A67CC74 020BBEA6 3B139B22 514A0879 8E3404DD
EF9519B3 CD3A431B 302B0A6D F25F1437 4FE1356D 6D51C245
E485B576 625E7EC6 F44C42E9 A637ED6B 0BFF5CB6 F406B7ED
EE386BFB 5A899FA5 AE9F2411 7C4B1FE6 49286651 ECE45B3D
C2007CB8 A163BF05 98DA4836 1C55D39A 69163FA8 FD24CF5F
83655D23 DCA3AD96 1C62F356 208552BB 9ED52907 7096966D
670C354E 4ABC9804 F1746C08 CA237327 FFFFFFFF FFFFFFFF
```

The primality of the number has been rigorously proven.

Group 3: An Elliptic Curve Group Definition

The elliptic curve is based on the Galois field $GF [2^{155}]$ with 2^{155} field elements. The irreducible polynomial for the field is $u^{155} + u^{62} + 1$. The equation for the elliptic curve is $y^2 + xy = x^3 + ax + b$, where x, y, a, b are elements of the field.

Elliptic curve parameters: $a = 0$ and $b = 471951$ in decimal and 7338F in hex.

Generator's points: $x = 123$ in decimal and *7B* in hex; $y = 456$ in decimal and *1C8* in hex.

The group order (the number of curve points) is:

4567192616659071619386556591434463519676923731б,

which is 12 times the prime

3805993847215893016155463826195386266397436443.

This prime has been rigorously proven. The generating point (*x, y*) has order 4 times the prime; the generator is the triple of some curve point.

IPsec and IKE v2 Identifiers

The way IPsec and IKEv2 define security-relevant information is by assigning specific protocol identifiers, and by describing how those protocols are used. Those identifiers are listed in the Internet Assigned Numbers Authority, (IANA). When negotiating security associations, the source and destination choose security protocols and cryptographic algorithms by using identifiers.

The following examples of IPsec and IKE v2 identifiers are presented to give an idea of how the identifiers are used, but it is not a substitute for reading the actual document specification:

IKEv2 Exchange Types, Part of the header (HDR)

Value	Type	Reference
34	IKE_SA_INIT	[RFC4306]
35	IKE_AUTH	[RFC4306]
36	CREATE_CHILD_SA	[RFC4306]
37	INFORMATIONAL	[RFC4306]

IKEv2 Security Protocol Identifiers

Protocol ID	Protocol	Reference
1	IKE	[RFC4306]
2	AH	[RFC4306]
3	ESP	[RFC4306]
4	FC_ESP_HEADER	[RFC4595]
5	FC_CT_AUTHENTICATION	[RFC4595]

Confidentiality Algorithm Identifiers

Number	Name	Reference
1	ENCR_DES_IV64	[RFC1827]
2	ENCR_DES	[RFC2405]
3	ENCR_3DES	[RFC2451]
4	ENCR_RC5	[RFC2451]
5	ENCR_IDEA	[RFC2451]
6	ENCR_CAST	[RFC2451]
7	ENCR_BLOWFISH	[RFC2451]
8	ENCR_3IDEA	[RFC2451]
9	ENCR_DES_IV32	[RFC4306]
11	ENCR_NULL	[RFC2410]
12	ENCR_AES_CBC	[RFC3602]
13	ENCR_AES_CTR	[RFC3686]
14	ENCR_AES-CCM_8	[RFC4309]
15	ENCR-AES-CCM_12	[RFC4309]
16	ENCR-AES-CCM_16	[RFC4309]
18	AES-GCM with a 8 octet ICV	[RFC4106]
19	AES-GCM with a 12 octet ICV	[RFC4106]
20	AES-GCM with a 16 octet ICV	[RFC4106]
21	ENCR_NULL_AUTH_AES_GMAC	[RFC4543]

Integrity Algorithm Identifiers

Number	Name	Reference
1	AUTH_HMAC_MD5_96	[RFC2403]
2	AUTH_HMAC_SHA1_96	[RFC2404]
3	AUTH_DES_MAC	[RFC4306]
4	AUTH_KPDK_MD5	[RFC1826]
5	AUTH_AES_XCBC_96	[RFC3566]
6	AUTH_HMAC_MD5_128	[RFC4595]
7	AUTH_HMAC_SHA1_160	[RFC4595]
8	AUTH_AES_CMAC_96	[RFC4494]
9	AUTH_AES_128_GMAC	[RFC4543]
10	AUTH_AES_192_GMAC	[RFC4543]
11	AUTH_AES_256_GMAC	[RFC4543]

IKEv2 Authentication Method

Value	Authentication Method	Reference
1	RSA Digital Signature	[RFC4306]
2	Shared Key Message Integrity Code	[RFC4306]
3	DSS Digital Signature	[RFC4306]
9	ECDSA with SHA-256 on the P-256 curve	
10	ECDSA with SHA-384 on the P-384 curve	
11	ECDSA with SHA-512 on the P-521 curve	

Pseudorandom Function

Number	Name	Reference
1	PRF_HMAC_MD5	[RFC2104]
2	PRF_HMAC_SHA1	[RFC2104]
3	PRF_HMAC_TIGER	[RFC2104]
4	PRF_AES128_CBC	[RFC4434]
8	PRF_AES128_CMAC	[RFC4615]

SA Life Type and SA Duration

The SA life type and SA duration specify the time-to-live for the overall security association. When the SA expires, all keys negotiated under the association (AH or ESP) must be renegotiated. The life type values are as follows:

Type	Value
Reserved	0
Seconds	1
Kilobytes	2

For a given life type, the value of the life duration attribute defines the actual length of the component lifetime—either a number of seconds, or a number of Kbytes that can be protected.

If unspecified, the default value shall be assumed to be 28,800 seconds (8 hours).

Encapsulation Mode

Type	Value
Reserved	0
Tunnel	1
Transport	2

Cryptographic Suites for IPsec

When IPsec is implemented in manual mode, there are many algorithms available, but two IPsec systems cannot interoperate unless they are using the same algorithms. The IPsec group proposed in RFC 4308 (Hoffman, 2005), "Cryptographic Suites for IPsec," uses two user interface suites (UI suites) that can cover typical configuration policies, and which can be used by system administrators to ease the burden of selecting among the many options in implementing IPsec systems. The suite concept is similar to TLS/SSL in which all parameters are listed and included in a single suite number/name. The initiator offers the responder one or more suites that it is able to support and lets the responder choose one of them. The two suites listed in RFC 4308 are for the use of IPsec in virtual private networks.

Suite VPN-A

IPsec:

- **Protocol:** Encapsulating Security Payload (ESP) [RFC2406]
- **ESP encryption:** TripleDES in CBC mode [RFC2451]
- **ESP integrity:** HMAC-SHA1-96 [RFC2404]

IKE and IKEv2:

- **Encryption:** TripleDES in CBC mode [RFC2451]
- **Pseudorandom function:** HMAC-SHA1 [RFC2104]
- **Integrity:** HMAC-SHA1-96 [RFC2404]
- **Diffie-Hellman group:** 1024-bit Modular Exponential (MODP) [RFC2409]

Suite VPN-B

IPsec:

- **Protocol:** ESP [RFC2406]
- **ESP encryption:** AES with 128-bit keys in CBC mode [AES-CBC]
- **ESP integrity:** AES-XCBC-MAC-96 [AES-XCBC-MAC]

IKE and IKEv2:

- **Encryption:** AES with 128-bit keys in CBC mode [AES-CBC]
- **Pseudorandom function:** AES-XCBC-PRF-128 [AES-XCBC-PRF-128]
- **Integrity:** AES-XCBC-MAC-96 [AES-XCBC-MAC]
- **Diffie-Hellman group:** 2048-bit MODP [RFC3526]

Summary

A virtual private network (VPN) creates a private network using the infrastructure of a public network such as the Internet. A VPN is able to establish a tunnel on layer 2 (data layer) and layer 3 (network layer) of the OSI model. At layer 2, VPNs establish tunneling protocols such as the layer 2 forwarding (L2F) protocol, the point-to-point tunneling protocol (PPTP), and the layer 2 tunneling protocol (L2TP). The only VPN protocol for layer 3 is the IPsec.

The L2F, PPTP, and L2TP protocols are strictly tunneling protocols. IPsec provides the encryption and authentication that the layer 2 tunneling protocols lack. IPsec actually provides all of the following security services: data origin authentication, access control, confidentiality (encryption), connectionless integrity, rejection of replayed packets (a form of partial sequence integrity), and limited traffic flow confidentiality.

One factor to consider when implementing IPsec is that it requires the installation of the client's software in each remote client machine. This is not a problem when a company has control of all of its networked computers, but when it wants to provide secure access to suppliers or vendors, then installing the client's software is not possible.

An important aspect of security is how to manage, exchange, transport, or wrap keys in a security association. In IPsec, it is necessary for hosts, source and destination, to agree to the mechanisms used to share the secret keys, as well as the keys that are used for authentication/integrity and encryption services. IKE v2, RFC 4306 (Hoffman, 2005) combines the security concepts of authentication and key management to provide a framework for transferring key and data authentication that is independent of the key generation technique, encryption algorithm, and authentication mechanism.

Learning Objectives Review

1. Encapsulation Security Payload (ESP) provides the same security services as the Authentication Header (AH), but in two modes instead of one. (T/F)
2. ESP and AH use the same authentication algorithms. (T/F)
3. IPsec is a network layer VPN technology, meaning it operates independently of the application(s) that may use it. (T/F)
4. Once an IPsec tunnel is negotiated via IKE, one-to-many connections of various types (Web, email, file transfer, VoIP) can flow over it, each destined for different servers behind the VPN gateway. (T/F)
5. A Security Protocol Identifier (SPI) indicates whether the security protocol is an Encapsulation Security Payload (ESP) or an Authentication Header Protocol (AH). (T/F)
6. In the tunnel mode, authentication is provided between a client and a corporate VPN device or between two VPN devices. (T/F)
7. A VPN provides security when corporate data is transmitted over a public network. (T/F)

8. Two security associations (one in each direction) are required to secure typical, bi-directional communication between two hosts, or between two security gateways. (T/F)
9. In VPN equipment, it is possible to specify that the lifetime of the negotiated key can be set up in seconds or in bits. Therefore:
    a. If the installer selects seconds, a new session key is exchanged after the default lifetime, 5 hours.
    b. If the installer selects bits, then every 5 megabytes a new session key is exchanged.
    c. A new key is generated after the customer-specified number of seconds, or bits, has passed by.
    d. A and B
10. IPsec is considered the best tunneling protocol for IP networks because it provides strong security services such as Encryption, Authentication, and Key management. (T/F)
11. An Encapsulated Security Payload (ESP) security mechanism provides:
    a. Integrity
    b. Authentication
    c. Confidentiality
    d. All of the above
12. An Authentication Header (AH) security mechanism does not provide:
    a. Integrity
    b. Authentication
    c. Confidentiality
    d. None of the above
13. Which of the following is not a component of IPsec?
    a. Authentication Header
    b. Internet Key Exchange
    c. Key Distribution Center
    d. Encapsulating Security Payload
14. Both ESP and AH security protocols support ___________ and _________ modes of operation.
15. The Authentication Header provides support for ________________ and for ______ ___________.
16. L2F, PPP, L2TP, and IPSEC can be used for encryption and key management in IP environments. (T/F)
17. In IPsec, the SA associated with a connection could be AH, ESP, or both. (T/F)
18. In IPsec, which traffic security protocol(s) is (are) used to encipher and which one(s) is (are) used to authenticate?

19. Are all VPNs secured? Explain.
20. What are Diffie-Hellman groups?

References

Frankel, S., & Herbert, H. (2003). *The AES-XCBC-MAC-96 algorithm and its use with IPsec* (RFC 3566). Internet Engineering Task Force (IETF). Retrieved June 28, 2007, from http://www.ietf.org/rfc/rfc3566.txt?number=3566

Gleeson, B., Lin A., Heinanen, J., Armitage, G., & Malis, A. (2000). *A framework for IP based virtual private networks* (RFC 2764). Internet Engineering Task Force (IETF). Retrieved June 28, 2007, from http://www.ietf.org/rfc/rfc2764.txt?number=2764

Hoffman, P. (2004). *The AES-XCBC-PRF-128 algorithm fo the internet key exchange protocol (IKE)* (RFC 3664). Internet Engineering Task Force (IETF). Retrieved June 28, 2007, from http://www.ietf.org/rfc/rfc3664.txt?number=3664

Hoffman, P. (2005). *Cryptographic suites for IPsec* (RFC 4308). Internet Engineering Task Force (IETF). Retrieved June 28, 2007, from http://www.ietf.org/rfc/rfc4308.txt?number=4308

Kaufman, C. (Ed.). (2005). *Internet key exchange (IKEv2)* (RFC 4306). Internet Engineering Task Force (IETF). Retrieved June 28, 2007, from http://www.ietf.org/rfc/rfc4306.txt?number=4306

Kent, S. (2005a). *IP Authentication header* (RFC 4302). Internet Engineering Task Force (IETF). Retrieved June 28, 2007, from http://www.ietf.org/rfc/rfc4302.txt?number=4302

Kent, S. (2005b). *IP Encapsulating security payload (ESP)* (RFC 4303). Internet Engineering Task Force (IETF). Retrieved June 28, 2007, from http://www.ietf.org/rfc/rfc4303.txt?number=4303

Kent, S., & Seo, K. (2005). *Security architecture for the Internet protocol* (RFC 4301). Internet Engineering Task Force (IETF). Retrieved June 28, 2007, from http://www.ietf.org/rfc/rfc4301.txt?number=4301

Kivinen, T., & Kojo, M. (2003). *More modular exponential (*MODP*) Diffie-Hellman Groups for IKE* (RFC 3526). Internet Engineering Task Force (IETF). Retrieved June 28, 2007, from http://www.ietf.org/rfc/rfc3526.txt?number=3526

Orman, H. (1998). *The OAKLEY key determination protocol* (RFC 2412). Internet Engineering Task Force (IETF). Retrieved June 28, 2007, from http://www.ietf.org/rfc/rfc2412.txt?number=2412

Schiller, J. (2005). *Cryptographic algorithms for use in the Internet key exchange version 2 (IKEv2)* (RFC 4307). Internet Engineering Task Force (IETF). Retrieved June 28, 2007, from http://www.ietf.org/rfc/rfc4307.txt?number=4307