

# Virtual private networks, Part 2

Presented by developerWorks, your source for great tutorials

[ibm.com/developerWorks](http://ibm.com/developerWorks)

---

## Table of Contents

If you're viewing this document online, you can click any of the topics below to link directly to that section.

<a href="#">1. Tutorial tips</a>	<a href="#">2</a>
<a href="#">2. Authentication Header (AH) considerations</a>	<a href="#">3</a>
<a href="#">3. Encapsulating Security Protocol</a>	<a href="#">5</a>
<a href="#">4. Key exchange</a>	<a href="#">10</a>
<a href="#">5. VPN implementations</a>	<a href="#">14</a>
<a href="#">6. Wrapup</a>	<a href="#">15</a>

## Section 1. Tutorial tips

### Should I take this tutorial?

This tutorial is aimed at technical folks who want to understand the overall workings of a virtual private network or VPN. It is a survey of the VPN field, not an in-depth analysis. However, knowledge of basic networking concepts is recommended.

[Part 1](#) started with a high-level VPN overview, proceeded to the technologies involved, and delved into the IPSec protocol; here in Part 2, we'll take a closer look at this technology and examine some VPN implementations of note.

---

### About the author

For questions about the content of this tutorial, please contact the author, Larry Loeb, at [larryloeb@prodigy.net](mailto:larryloeb@prodigy.net).

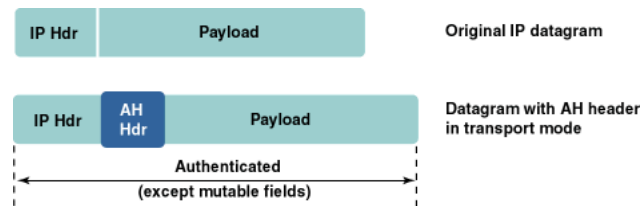
Larry Loeb has written for many of the last century's major "dead tree" computer magazines, having been -- among other things -- a consulting editor for *BYTE* magazine and senior editor for the launch of *WebWeek*. He's been online since uucp "bang" addressing (where the world existed relative to !decvax), serving as editor of the *Macintosh Exchange* on BIX, and the *VARBusiness Exchange*. He's also written a book on the Secure Electronic Transaction Internet protocol. His first Mac had 128K of memory. His first 1130 had 4K, as did his first 1401.

## Section 2. Authentication Header (AH) considerations

### AH use in transport mode

In this mode, the original IP datagram is taken and the AH header is inserted right after the IP header. If the datagram already has an IPSec header, then the AH header is inserted before any of those. Transport mode is used by hosts, not by gateways. In fact, gateways are not required to support transport mode. The advantage of the transport mode is it requires less processing overhead. The disadvantage is that the mutable fields are not authenticated.

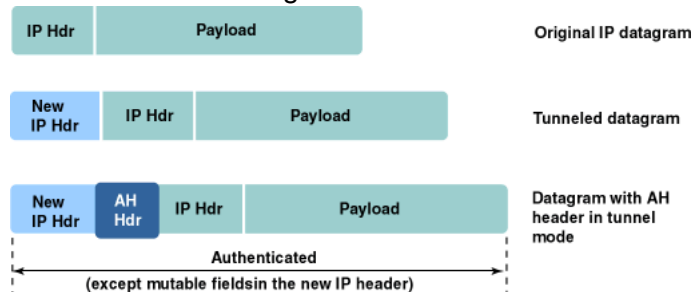
Figure 1



### AH use in tunnel mode

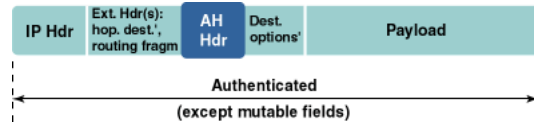
In tunnel mode, the IP datagram is the payload of a new IP datagram. This mode is used whenever either end of a security association is a gateway. So, if a connection is made between two firewalls, the tunnel mode is always used.

In tunnel mode the outer headers' IP addresses do not need to be the same as the inner headers' addresses. For example, two security gateways can operate an AH tunnel, which is used to authenticate all traffic between the networks they connect together. Hosts are not required to support tunnel mode, but often they do. The advantages of the tunnel mode are total protection of the encapsulated IP datagram and the possibility of using private addresses. However, there is an extra processing overhead associated with this mode. Figure 2



## IPv6 tunnel considerations

In IPv6, AH is considered an end-to-end payload and it appears after hop-by-hop, routing, and fragmentation extension headers. The destination options extension header(s) could appear either before or after the AH header. In Figure 3, the extension headers marked with an asterisk have variable positions, if they are present. Figure 3



## Section 3. Encapsulating Security Protocol

### ESP overview

Encapsulating Security Protocol (ESP) provides integrity check, authentication, and encryption for IP datagrams. It operates on a per-packet basis. The desired services to be used are selectable upon establishment of the security association (SA). However, some restrictions apply: Integrity check and authentication go together. Replay protection is selectable only with integrity check and authentication. Replay protection can be selected only by the receiver. Encryption is selectable independent of any other services. If encryption is enabled, then integrity check and authentication should be turned on as well. If only encryption is used, intruders could forge packets in order to mount a cryptanalytic attack.

---

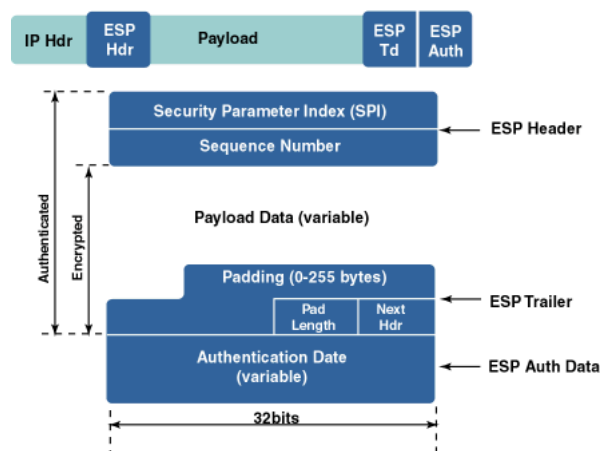
### ESP specifics

ESP is identified by protocol number 50. The protocol header (IPv4, IPv6, or Extension) immediately preceding the AH header will contain this value in its Protocol (IPv4) or Next Header (IPv6, Extension) field. ESP will only work with non-fragmented IP packets. However, an IP packet with ESP applied can be fragmented by intermediate routers. So, the destination must first reassemble the packet and then use ESP. If an IP packet appears to be a fragment (the offset field is non-zero, or the More Fragments bit is set), it is discarded by ESP. If both encryption and the authentication with integrity check are active in the ESP processing, the receiver will first authenticate the packet. If this step is successful, ESP then proceeds with decryption. This saves computing resources and reduces vulnerability to denial-of-service attacks.

---

## ESP packet format

The format of the ESP packet is more complicated than that of the AH packet. There is an ESP header, an ESP trailer, and ESP authentication data. The payload is located (encapsulated) between the header and the trailer, which gives the protocol its name. Figure 4



### Packet fields: Security Parameter Index, Sequence Number, Payload Data

The ESP packet contains the following fields:

**Security Parameter Index (SPI)** This field is 32 bits in length, and is defined the same as it was in the AH discussion (see [Part 1](#)).

**Sequence Number** This 32-bit field is an increasing counter, and again follows the AH definitions. It is used to prevent replay attacks.

**Payload Data** This field is mandatory. It consists of a variable number of bytes of data described by the Next Header field. PD is encrypted with the cryptographic algorithm selected during SA establishment. If the algorithm requires initialization vectors, they will be included here as well.

The ESP specification requires support for the DES algorithm in CBC mode (DES-CBC transform). Other encryption algorithms are also supported, such as triple-DES.

## Packet fields: Padding, Pad Length, Next Header

**Padding** Most encryption algorithms require that the input data must be an integral number of blocks. The resulting ciphertext (which includes the Padding, Pad Length, and Next Header fields) have to terminate on a 4-byte boundary. This is so that the Next Header field is right aligned. Padding is an optional field, so whatever cipher is used, it has a field to insure correct alignment.

**Pad Length** This 8-bit field contains the number of the preceding padding bytes. It is always present, and the value of 0 indicates no padding.

**Next Header** The Next Header is an 8-bit mandatory field that shows the data type carried in the payload. The data type is one of those on the IP Protocol Numbers List.

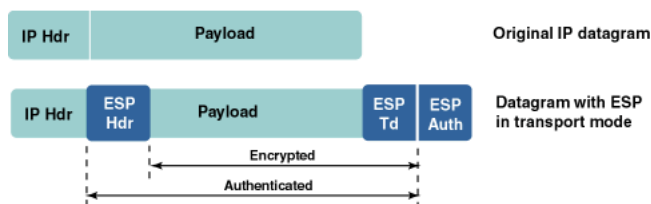
---

## Packet fields: Authentication Data, Transport Mode

**Authentication Data** This optional field is variable in length, and is included only when integrity checks and authentication are active. The AD is calculated from the SPI to the Next Header field inclusive. The ESP specifications require that two authentication algorithms be supported: HMAC with MD5 and HMAC with SHA-1. Remember that the IP header is not covered in the AD.

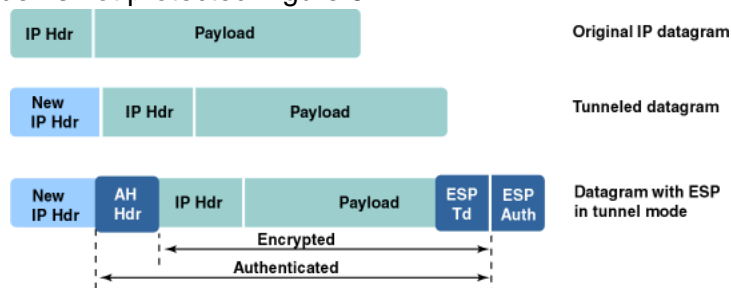
**Transport Mode** Here the ESP header is inserted immediately after the IP header. If the datagram already has IPsec header(s), then the ESP header is inserted before any of those. The ESP trailer and the optional authentication data are appended to the payload. In transport mode, ESP neither authenticates nor encrypts the IP header. But this mode has a low computational overhead. Like AH, transport mode is for hosts. Gateways do not even have to support this mode.

Figure 5



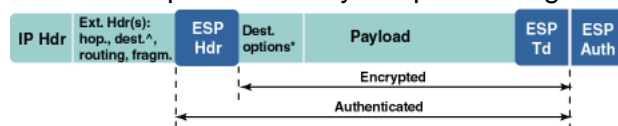
## Packet fields: Tunnel Mode

Tunnel Mode Here, a new IP packet is made with a new IP header. ESP/Transport is then applied and the original datagram is the payload for the new packet. If authentication and encryption have been initiated, the packet is protected. However, the new IP header is not protected. Figure 6



## IPv6 and ESP

In IPv6, ESP is an end-to-end payload and it appears after hop-by-hop, routing, and fragmentation extension headers. The destination options extension header(s) could appear either before or after the AH header. Those headers marked with an asterisk (\*) in Figure 7 may have a variable position if they are present. Figure 7



## Why ESP and AH?

The question might be asked, "Why would AH be supported in IPsec if the more encompassing ESP is available?" There are two main reasons. First, ESP requires "strong" cryptographic methods. Even though the U.S. export policy has been recently liberalized, these policies are not universal. Depending on the local governmental climate, strong crypto may still be problematic. However, because AH is for authentication, it can be used and transmitted globally. Second, one may need only authentication in a particular setting. AH uses less processing, and thus will be desirable for performance reasons.

## Combined use of AH and ESP I

As mentioned, AH and ESP can be applied alone or used in combination. Interestingly, AH and ESP SAs don't have to have the identical endpoints. If this is to be the case, at least one level of tunneling must be incorporated into the SA. There are two approaches for an SA bundle creation: Transport adjacency: Both security protocols are applied in transport mode to the same IP datagram. This method is practical for only one level of combination. Iterated (nested) tunneling: The security protocols are applied in tunnel mode in sequence. After each application, a new IP datagram is created and the next protocol is applied to it. This method has no limit in the nesting levels. However, using more than three levels of nesting has proven to be impractical. It's possible to combine the approaches. For example, an IP packet with transport adjacency IPsec headers can be sent through a nested tunnel.

---

## Combined use of AH and ESP II

Upon the receipt of a packet with both protocol headers, the processing sequence should be authentication followed by decryption. Why decrypt if you are not certain of the origin? So, the sender should first apply ESP and then AH to the outbound traffic. In fact, this sequence is an explicit requirement for transport mode IPsec processing. When using both ESP and AH, one must consider whether ESP authentication should be turned on since AH authenticates the packet anyway. The answer depends on the relative extent of the SAs used. Turning ESP authentication on makes sense when the ESP SA extends beyond the AH SA. Here, ESP can avoid spoofing attacks in the intranet. In general, the transport mode is used between the endpoints of a connection, and tunnel mode is used between two machines when at least one of them is a gateway.

## Section 4. Key exchange

### Internet Key Exchange I

As has been noted earlier in this tutorial, ISAKMP provides a framework for authentication and key exchange but does not define them. ISAKMP is designed to support many different key exchanges. "Oakley" is a description of a series of key exchanges that it calls *modes*. The protocol details the services provided by each one, namely "perfect forward secrecy" for keys, identity protection, and authentication. SKEME, first described in the "IEEE Proceedings of the 1996 Symposium on Network and Distributed Systems Security" is a key exchange technique that provides anonymity, some repudiability, and quick key refreshment. The IKE RFC (2049) describes a hybrid that uses elements of both in conjunction with ISAKMP to obtain authenticated keying material for use with ISAKMP, and for other security associations such as AH and ESP.

---

### Internet Key Exchange II

IKE supports client negotiation. Client mode is where the negotiating parties are not the endpoints for which security association negotiation is taking place. When used in client mode, the identities of the end parties remain hidden. Before going into the RFC 2049 details, it must be understood that IKE is not yet fully deployed on the Internet because the total system is still under development. Parts of it (like 2049) seem relatively stable, but other parts (like the specific hash function to be used in IKE) are still in the draft stages. Also, there is not a distributed PKI yet that is available to all. A VPN can use the overall framework of IKE, but must look at a specific implementation to make sure that it will perform as needed.

---

### Internet Key Exchange III

A central idea of Oakley's PFS is that compromise of a single key will permit access to only data protected by a single key. For PFS to exist, the key used to protect transmission of data must not be used to derive any additional keys, and if the key used to protect transmission of data was derived from some other keying material, that material shall not be used to derive any more keys. This assures the "freshness" of a key. While Oakley defines "modes," ISAKMP defines "phases." IKE presents different exchanges as modes that operate in one of two phases: Phase 1 is where the two ISAKMP peers establish a secure, authenticated channel with which to communicate (the SA process). "Main Mode" and "Aggressive Mode" (we'll get to these later), each accomplish a phase 1 exchange. "Main Mode" and "Aggressive Mode" are only used in phase 1. Phase 2 is where SAs are negotiated on behalf of IPSec or another service that needs key material. "Quick Mode" accomplishes a phase 2 exchange.

---

## Key exchange methods

We've noted the two main methods of key exchange: Main Mode and Aggressive Mode. Each generates authenticated keying material from a Diffie-Hellman exchange. Main Mode must be implemented; Aggressive Mode should also be implemented. Quick Mode must be implemented to generate fresh keying material as well as negotiating non-ISAKMP security services. Also, exchange types must now be switched in the middle of an exchange. The SA payload must precede all other payloads in a phase 1 exchange. Main Mode follows the ISAKMP Identity Protect Exchange: The first two messages negotiate policy; the next two exchange Diffie-Hellman public values and other data necessary for the exchange; and the last two messages authenticate the Diffie-Hellman Exchange.

---

## Aggressive Mode

The first two messages in Aggressive Mode negotiate policy, exchange Diffie-Hellman public values and ancillary data necessary for the exchange, and identities. In addition, the second message authenticates the responder. The third message authenticates the initiator and provides a proof of participation in the exchange. The final message is not to be sent under protection of the SA. This allows each party to postpone exponentiation, if desired, until negotiation of this exchange is complete. Security Association negotiation is limited with Aggressive Mode. For example, the group in which the Diffie-Hellman exchange is performed cannot be negotiated. Differing authentication methods may further constrain attribute negotiation. Similarly, authentication with public key encryption cannot be negotiated. If there are situations where the richer attribute negotiation capabilities of IKE are necessary, Main Mode may be required.

---

## SA negotiation

The SA negotiation takes the form of Transform Payload(s) encapsulated in Proposal Payload(s), which are further encapsulated in SA payload(s). If multiple offers are being made for phase 1 exchanges in Main Mode and Aggressive Mode, these offers must take the form of multiple Transform Payloads for a single Proposal Payload in a single SA payload. There is no inherent limit on the number of offers the initiator may send to the responder. Some implementations may choose to limit the offers to increase performance, however. In the SA negotiation, initiators present offers for potential SAs to responders. Responders cannot modify attributes of an offer, except for attribute encoding. If the initiator of an exchange finds that (1) attribute values have changed, or (2) attributes have been added or deleted from the offer made, then the changed response must be rejected.

---

## Authentication I

There are four different authentication methods allowed in either Main or Aggressive Mode. These methods are: digital signature, two forms of authentication with public key encryption, or a pre-shared key. For an authentication with digital signatures, message hashes are signed and verified; for authentication with either public key encryption or pre-shared keys, the hashes directly authenticate the exchange. When using public keys for authentication, the Phase 1 exchange can be accomplished either by using signatures or by using public key encryption. With signatures, the exchange is authenticated by signing a mutually obtainable hash. RSA signatures must be encoded as a private key encryption in PKCS #1 format and not as a signature in PKCS #1 format. This is because the latter includes the Object Identifier (OID) of the hash algorithm, and that OID is invalid in IPsec. Also, authentication with public key encryption allows for identity protection with Aggressive Mode.

---

## Authentication II

Another kind of authentication (pre-shared key) works with Main Mode. The key can only be identified by the IP address of the peers since the hash is computed before a relevant address field. Aggressive Mode will allow a wider range of identifiers of the pre-shared secret. In addition, Aggressive Mode allows two parties to maintain multiple, different pre-shared keys and identify the correct one for a particular exchange. Quick Mode is essentially an SA negotiation and an exchange of nonces that provides replay protection. The nonces are used to generate fresh key material and prevent replay attacks. An optional Key Exchange payload can be exchanged to allow for an additional Diffie-Hellman exchange and exponentiation per Quick Mode. Using the key exchange payload with Quick Mode is optional, but must be supported.

---

## Perfect Forward Secrecy

The Perfect Forward Secrecy protocol masks identities of both the ISAKMP negotiating peer and, if applicable, the identities for whom the peers are negotiating. This protects the IP addresses of an intranet, for example. To provide Perfect Forward Secrecy of both keys and all identities, the two parties involved have the following exchanges: A Main Mode Exchange to protect the identities of the ISAKMP peers. This establishes an ISAKMP SA. A Quick Mode Exchange to negotiate other security protocol protection. This will establish an SA on each end for PFS. Delete the ISAKMP SA and its associated state. To provide Perfect Forward Secrecy of just the keys of a non-ISAKMP security association, it is not necessary to do a phase 1 exchange if an ISAKMP SA exists between the two peers. A single Quick Mode in which the optional KE payload is passed and an additional Diffie-Hellman exchange is performed, is all that is required. At this point, the state derived from this Quick Mode must be deleted from the ISAKMP SA.

---

## Some useful optimizations I

In Quick Mode negotiating a range of SAs will speed up the "re-keying." When one peer feels it is time to change SAs they simply use the next one within the stated range. A range of SAs can be established by negotiating multiple SAs (identical attributes, different SPIs) with one Quick Mode. Establishing SAs with peers before they are needed ensures there will be no delays due to key management before initial data transmission. Multiple negotiations are performed, and those not immediately used are cached. To make things even faster, if ISAKMP is alerted that a SA will soon be needed (say, to replace an expiring SA) then it can establish the new SA before that new SA is needed.

---

## Some useful optimizations II

The ISAKMP specification describes conditions in which one party of the protocol might inform the other party of some activity -- either deletion of a security association or in response to some error in the protocol such as if a signature verification failed or a payload failed to decrypt. These informational exchanges should not be responded to under any circumstances. It's too easy to get into a tight message loop where the peers notify each other about the messages that are being passed back and forth. Not worth it. IKE exchanges maintain running initialization vectors (IV) where the last ciphertext block of the last message is the IV for the next message. To prevent retransmissions (or perhaps forged messages with valid cookies) from causing exchanges to get out of sync, IKE implementations cannot update their running IV until the decrypted message (1) passes some basic sanity check, and (2) has been determined to actually advance the IKE state machine, which means the message is not a retransmission.

## Section 5. VPN implementations

### Products of note: Overview

ICSA Labs (see the Resources section) has certified 26 gateways/switchers from 16 different vendors as being compliant with IPsec and being interoperable with each other. Also, there are four client-side software packages it has certified that will operate with the certified gateways under static IP conditions. The use of static IP conditions implies that the dynamic addresses assigned through DHCP or NAT might not always work correctly through heterogeneous gateways. It's up to the product specifier to make sure that any impact on the proposed usage of these products is understood.

---

### Products of note: AIX

AIX from IBM incorporates an IPsec VPN in the 4.3.3 distribution. The included client is on the ICSA list. It can be managed graphically, and has good documentation. It is a "pure" IPsec implementation that does not use L2TP or PPTP, which are available on other IBM networking products. The IKE in AIX uses only RSA signatures or shared keys. The ability to use X.509 certificates (and perhaps LDAP) should probably show up in version 5.0.

---

### Products of note: FreeS/WAN

Open source VPNs are available, but are totally unsupported by any central responsible party. FreeS/WAN is probably the leading product in this niche. It runs under Linux. There is fairly detailed documentation that goes along with it, and you can connect it with other IPsec solutions being used elsewhere in the system. The IKE module uses RSA signatures or shared keys only. (Remember, it was mentioned earlier that X.509 certs aren't being used much yet -- there's a bit of a chicken-and-egg situation currently going on.)

---

### Products of note: BSD Linux, the IKE module

BSD Linux contains one of the most full-featured open source IPsec VPNs. It is very interoperable and has a reliable reputation in the field. The VPN's documentation, however, is weak. The IKE module (racoon) can be extended to accept X.509 certs, although it does not yet validate the cert to the issuing server. It also uses 3DES (triple-DES) for the encryption, which brings it up to current NIST standards. Additionally, a packet filter is built into BSD that can be configured to run as a firewall in conjunction with the VPN. Interestingly, this VPN implementation will run on the Macintosh##i°# OS X because OS X uses BSD networking.

## Section 6. Wrapup

### Summary

A VPN can be an implementation problem due to the interactions possible over a heterogeneous network. Authentication via IPSec is currently limited to shared secrets or RSA signatures, but in the future it will need to migrate to a workable PKI certificate system. Many commercial VPN solutions exist, as well as open source implementations. The lack of support and documentation for the open source efforts might limit their usefulness in mission-critical systems, but might not affect systems that have small numbers of users and defined hardware.

---

### Resources

Karen Monkhouse's overview of VPN technology, "[Business Taps the Internet with Virtual Private Networks](#)", includes several case studies. Visit MIT's [ISAKMP Distribution Page](#). RFC 2003, "[IP encapsulation within IP](#)", shows you how an IP datagram can be encapsulated within another IP datagram. Check out RFC 2401, "[Security Architecture for the Internet Protocol](#)" by Stephen Kent and Randall Atkinson. [Winn Schwartau's Network World column](#) tells you what you need to know about VPNs. RSA Security's Web site features the white paper "[Implementing A Secure Virtual Private Network](#)". Get the latest [IBM security news](#). Find out how [IBM's Managed Security Services](#) can help you to identify and solve your real-time security risks by using a proven continuous management process. ICSA Labs (<http://www.icsalabs.com/>), a division of the TruSecure Corp., is a central authority for research, intelligence, and product certification for the security industry. For more on AIX, IBM's scalable UNIX platform, check out <http://www-1.ibm.com/servers/aix/>. Linux FreeS/WAN is an implementation of IPSEC and IKE for Linux. To find out more, head for <http://www.freeswan.org/>.

---

### Your feedback

Please let us know whether this tutorial was helpful to you and how we could make it better. We'd also like to hear about other tutorial topics you'd like to see covered. Thanks!

For questions about the content of this tutorial, please contact the author, Larry Loeb, at [larryloeb@prodigy.net](mailto:larryloeb@prodigy.net).

---

---

### Colophon

This tutorial was written entirely in XML, using the developerWorks Toot-O-Matic tutorial generator. The Toot-O-Matic tool is a short Java program that uses XSLT stylesheets to

convert the XML source into a number of HTML pages, a zip file, JPEG heading graphics, and two PDF files. Our ability to generate multiple text and binary formats from a single source file illustrates the power and flexibility of XML.