Chapter 21

IPsec Overview

Companies (or anyone) exchanging electronic data between the home office and field offices want secure communication assurances. Leased lines, dedicated to the lessor, provide secure communications, but this approach is too expensive and much less flexible than Internet communications. A virtual private network (VPN) based on Internet Protocol Security (IPsec) is the current commercial choice for secure Internet communications.

Internet security is driving IPsec.

Business to business (B2B) electronic communication is becoming a necessity for companies' survival. For example, some hospitals permit their suppliers access to their network and internal databases. The supplier queries the hospital database to analyze levels of supplies and then is able to deliver those supplies that are needed. Obviously, this arrangement is efficient. But the hospital needs to protect some parts of its internal network, such as patients' medical records. Similarly, computer road warriors want assurances that when they log on to a home office computer server from a hotel room, all the data they exchange with the home office is secure.

In the not-too-distant future, it's predicted that most Internet users will control their bank accounts, health insurance, and perhaps even home appliances through the Internet.

Enhanced Security

IPsec can authenticate any data packet that enters and encrypt any data packet that leaves.

IPsec (sometimes spelled IPSec) offers authentication, confidentiality, integrity, access control, protection against replay attacks, and limited protection against traffic flow analysis. In brief, an IPsec-enabled computer can authenticate any data packet that enters and encrypt any data packet that leaves.

In Chapter 20 you saw how Alice and Bob use SSL/TLS to secure Internet transactions. Secure e-mail and SSL/TLS are application programs, and they usually require that the user request cryptographic services; the use of cryptography is not automatically the default.

IPsec is completely transparent to the user.

IPsec, in contrast, operates under the application level, transparent to the user. It empowers an IPsec administrator[1] to provide cryptographic protections to all incoming and outgoing Internet data transfers. This means that an IPsec-enabled computer automatically protects e-mail, Web browsing, file transfers—any electronic communication between itself and another IPsec-enabled computer. IPsec automatically negotiates cryptographic protections with another IPsec-enabled computer that has acceptable credentials. If the other computer is not IPsec-enabled, IPsec can allow or disallow communication in a way that's transparent to the user. Microsoft has already embedded many IPsec features into Windows 2000.

IPsec negotiations are secret.

Another significant IPsec security feature is that cryptographic protections, such as the choice of cipher method, can be secretly negotiated. In contrast, SSL/TLS protections are negotiated with plaintext messages. (In Chapter 20, Bob sends a plaintext message to Alice suggesting a cipher method; Alice responds with a plaintext message.)

In this chapter we present an overview of IPsec and explain some of the benefits it offers to HxMel employee Bob as he connects through the Internet to Alice at AliceDotComStocks.[2, 3] As we discuss IPsec in this chapter, we use the symbols shown in Figure 21-1.

Key Management

IPsec-compliant systems[4] must support manual distribution and automated negotiation of secret keys.

1. Or knowledgeable user.
2. As in Chapter 20, where appropriate we'll abbreviate AliceDotComStocks as Alice.
3. Technical Note: SSL and TLS are implemented above the transport layer at the application layer. IPsec, in contrast, is implemented below the transport layer. The base standards document suggests three ways to implement IPsec:
   1. Integration into Internet Protocol (IP); changes to the IP source code are required.
   2. Under IP, between IP and native drivers; no changes to IP source code are required. This is referred to as a bump in the IP stack (BITS).
   3. Outboard crypto processor, referred to as bump in the wire (BITW).
   In any case, applications don't need to know it's there.
4. That is, IPsec-compliant systems that follow the IETF standards.

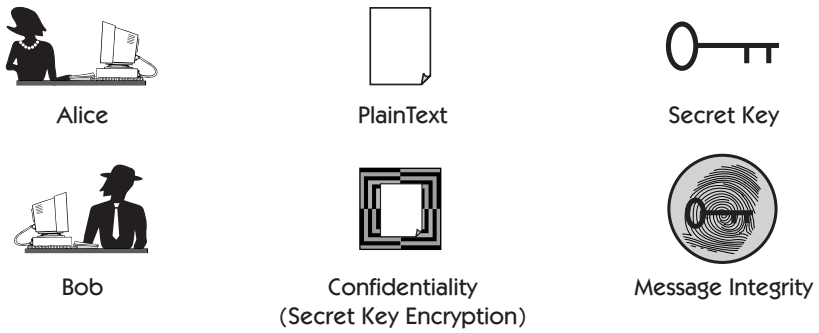

Figure 21-1 Symbols used in this chapter.

Manual Distribution

Manual distribution means that a controlling authority acts as a key distribution center (KDC, discussed in Chapter 8) and manually distributes secret keys. Although manual distribution is the simplest form of key management, it has the same problems as a KDC—for example, difficulty in changing secret keys. Manual key distribution is suitable for small IPsec installations.

Automated Distribution

Automated key management is required for any system except a small user group. Automated negotiation makes and distributes secret keys as needed; arguably, it's the most complex and controversial part of IPsec. In addition, automated key management provides protections not available in manual management—for example, anti-replay protection. In the following overview of user authentication and key agreement, we examine IPsec's default automated key management system.

IPsec splits into two parts: key management and bulk data encryption.

In this overview, we treat IPsec as consisting of two parts (see Figure 21-2). In the first part, Alice and Bob negotiate cryptographic parameters and assurances, complete authentication, and agree on shared secret keys. The second part provides bulk data encryption confidentiality and message integrity.

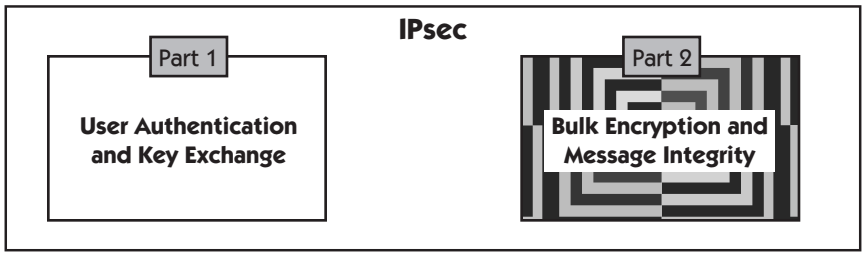

Figure 21-2 IPsec overview. Portions of part 1 communications are completed with plaintext messages; part 2 communications consist entirely of encrypted transmissions.

IPsec Part 1: User Authentication and Key Exchange Using IKE

Although the IPsec standard allows more than one automated key management technique, Internet Key Exchange (IKE) is the default IPsec key exchange protocol. Most IPsec vendors have implemented a version of IKE in their products.

SSL/TLS and IPsec Key Agreement

SSL has one set of parameters, one secret exchange in one phase. IKE has two sets of parameters, two secret exchanges in two phases.

In SSL/TLS, Alice and Bob exchange one secret and negotiate one set of cryptographic parameters. Cryptographic parameters include choices such as bulk encryption method and message digest method. SSL/TLS parameter negotiations are completed with one round-trip message: one message from Bob to Alice and one message from Alice to Bob.

IKE is more complicated than SSL/TLS key management. In IKE, Alice and Bob exchange two secrets and negotiate two sets of cryptographic parameters; each secret is associated with a set of parameters. You'll see how, compared with a single key exchange and a single set of cryptographic parameters, the use of two secret key exchanges and two sets of cryptographic parameters adds security and speed.

Security Association

Definition: security association (SA)

One secret key together with one set of cryptographic parameters is called a *security association* (SA). SAs are very similar to SSL/TLS cipher suites (discussed in Chapter 20); SAs contain shared secret keys, the names of cryptographic methods that Alice and Bob use for encryption and authentication, and other parameters.

Phases

IKE has two phases. Each phase makes an SA.

IKE establishes two SAs in two-phase negotiations between Alice and Bob (see Figure 21-3). Phase 1 exchanges are mostly plaintext (unencrypted) messages. Phase 2 exchanges are all encrypted messages. Phase 1 makes SA-1; phase 2 makes SA-2. SA-1 parameters are used to encrypt and authenticate phase 2 messages. SA-2 parameters are used to encrypt and/or authenticate all part 2 (bulk data encryption) messages.

Phase 1: Key Agreement and Authentication

As with SSL/TLS, at the start of IKE phase 1 Alice and Bob must communicate without encryption because they haven't agreed on cryptographic parameters and cryptographic keys. So IKE phase 1 consists mostly of plaintext message exchanges to negotiate cryptographic parameters and shared secret keys (see Figure 21-4).

IKE uses Diffie-Hellman key agreement.

In phase 1, shared secret keys are established using Diffie-Hellman key agreement (see Chapter 9 and Appendix A); Bob authenticates Alice and vice versa. Phase 1 can complete with three messages (*aggressive* mode) or six messages (*main* mode). Using three messages completes faster, but using six messages offers additional features, such as some denial of service protection.[5]

Phase 1 makes an authenticated secure channel between Alice and Bob.

At the completion of phase 1, Alice and Bob have set up an authenticated secure channel between them. With their first shared secret they derive three secrets: an encryption key, an authentication key, and an additional secret value. The keys are used to encrypt and authenticate all the phase 2 messages; the additional shared secret value is used to derive the second set of secret keys.

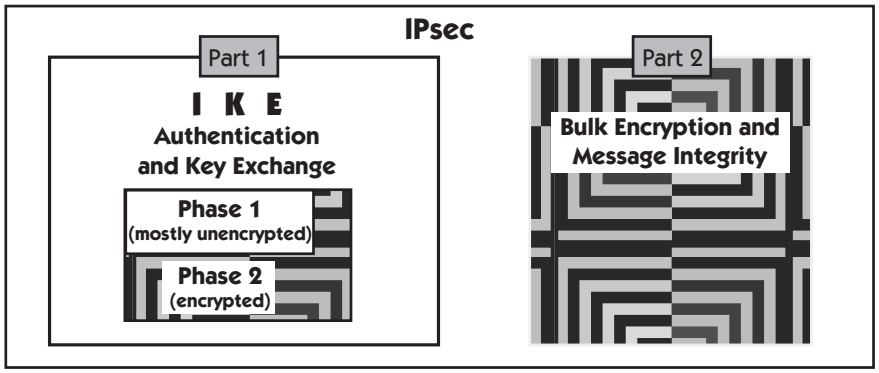

Figure 21-3 IPsec showing IKE phases.

5. See "Clogging Attack" in Appendix B.

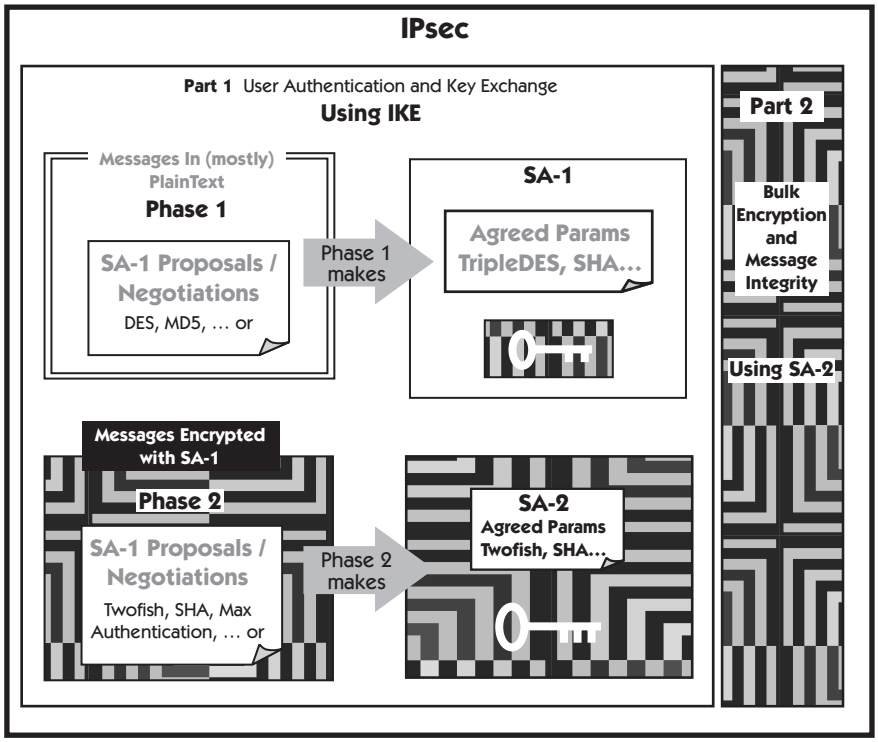

Figure 21-4 IKE phase 1 and phase 2 negotiations. In phase 1 they agree to use the parameters in SA-1 (Triple DES, SHA-1, …). All phase 2 negotiations are secured with SA-1 parameters. In phase 2 they agree on SA-2 (Twofish, SHA-1, …). SA-2 parameters are used in IPsec part 2.

Phase 2: Setting Up Bulk Encryption Parameters

Definition: quick mode

All phase 2 messages are encrypted and authenticated with the SA-1 cryptographic methods and shared secret key. Phase 2 always completes in three messages. Phase 2 doesn't have any time-consuming public key operations and quickly completes; it is called *quick mode*. In phase 2, Alice and Bob negotiate cryptographic parameters used for bulk data encryption and calculate their second set of shared secret keys (see Figure 21-4).

The second set of shared secret keys is calculated from the additional secret value made in phase 1 along with new random numbers Alice and Bob inject into the process.

IKE Nomenclature

Definitions: IKE-SA, IPsec-SA

As mentioned earlier, in each of IKE's two phases Alice and Bob negotiate and agree on an SA. We took some liberty and renamed the SAs. The first SA negotiated in IKE phase 1 is actually called IKE SA (our SA-1). Then IKE phase 2 actually produces IPsec SA (our SA-2). Because IPsec is used to name the whole process and IKE is the name of the key exchange protocol, the names IKE SA and IPsec SA can be confusing. So for simplicity, we continue to refer to the IKE SA as SA-1 and the IPsec SA as SA-2 (see Table 21-1). As far as we know, these names are used only in this book.

Benefits of Two-Phase Key Exchange

IPsec's two-phase key exchange is designed to negotiate new bulk encryption keys quickly and securely as well as facilitate changes in bulk encryption methods.

Changing Bulk Data Encryption Keys

Secret keys age (they get used up).

Secret keys age; each time they're used, BlackHat gets more clues to use for cryptanalysis. After a while, you should replace old secret keys with new ones.

If either Alice or Bob decides that a shared secret key is no longer secure, two-phase key exchange allows them to securely and quickly change the secret key by performing another phase 2 (quick mode) exchange and making a new SA-2.

Phase 2 can be used to quickly negotiate new secret keys.

Phase 2 is fast because it uses secret key encryption rather than public key encryption. As noted in previous chapters, secret key encryption is much faster than (and is as secure as) public key encryption. Appendix B has more on IKE phases and options.

Definition: SA lifetime

It's so fast and easy to share a new SA-2 that a new SA-2 is given a *lifetime*, expressed either as a given amount of time or as a given amount of plaintext encrypted. After its lifetime expires, Alice's and Bob's computers make and share

Table 21-1 Simplification of IKE nomenclature in this book.

Conventional IKE Names	Names Used in This Book
IKE SA	SA-1
IPsec SA	SA-2

a new SA-2. IPsec handles this automatically; Alice and Bob may not even know it's happening. In contrast, SSL/TLS uses much slower public key encryption to agree on a new bulk data secret key. Of course, if the secret key Alice and Bob agree on in IPsec phase 1 is compromised, they must also begin a new phase 1.[6]

Creating Bulk Encryption Keys for Separate Applications

Another benefit of two-phase exchange is that you can use a single SA-1 to create many SA-2s. For example, Figure 21-5 shows Alice and Bob using their SA-1 to create an SA-2 for encrypting database files and another SA-2 for encrypting e-mail files. Note that the "database" SA-2 uses different cryptographic parameters from those of the "e-mail" SA-2.

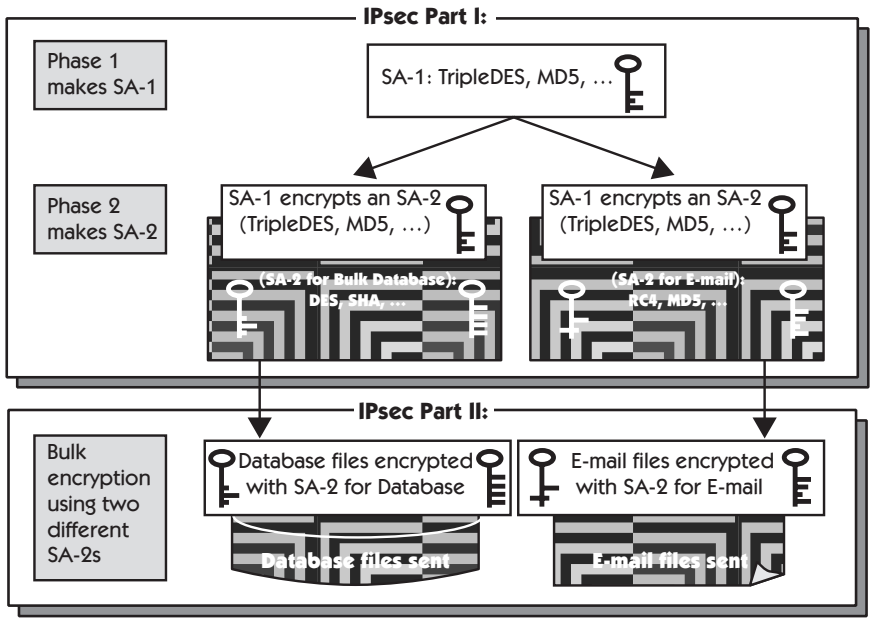

Figure 21-5 IKE can create many SA-2s. Here, IKE creates two SA-2s, one for each separate application. Note that each SA-2 has different parameters and keys.

6. After an agreed-on time period, phase 1 keys also expire and must be renegotiated.

IPsec Part 2: Bulk Data Confidentiality and Integrity for Message or File Transport

In IPsec part 2, Alice and Bob exchange encrypted messages using the parameters and secret keys (SA-2) made in IKE phase 2. After Alice and Bob agree on SA-2, IKE hibernates until another key negotiation or new cryptographic parameters are needed. Figure 21-6 shows some of IPsec's bulk data encryption features.

A particular SA-2 is used only for inbound or outbound messages and not both.

IPsec requires that an SA-2 be used in only one direction: either for inbound or outbound messages and not both. This means that IPsec requires Alice to have an SA-2 that is used only for her outbound messages. Bob's inbound SA-2 must have the identical cryptographic parameters and keys so that he can decrypt Alice's messages. Similarly, IPsec requires Bob to have an SA-2 that is used only for his outbound messages; and again, Alice's inbound SA-2 must have the identical copy of his cryptographic parameters and keys so that she can decrypt his messages.[7]

In Figure 21-7, Alice's outbound cryptographic parameters and keys are equivalent to Bob's inbound and vice versa. Although, in our example, the only differences between Alice's inbound and outbound SAs are the secret keys, IPsec allows more options. For example, although Alice's outbound traffic used DES, in theory Bob's outbound could use Triple DES. Obviously, for this to work, both Alice and Bob must accommodate DES and Triple DES.

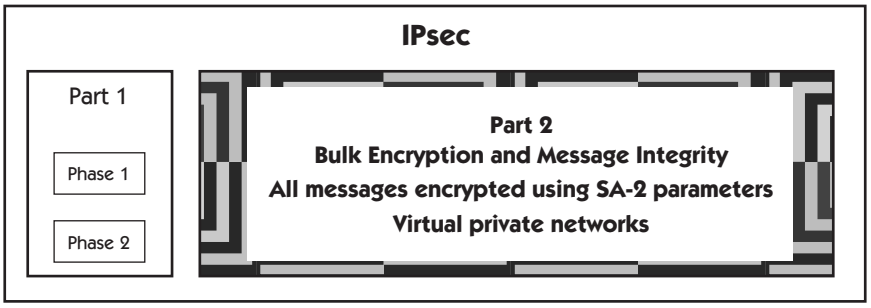

Figure 21-6 Overview of IPsec part 2.

7. Although SSL/TLS uses different cryptographic keys for outgoing and incoming traffic, the cryptographic methods are the same.

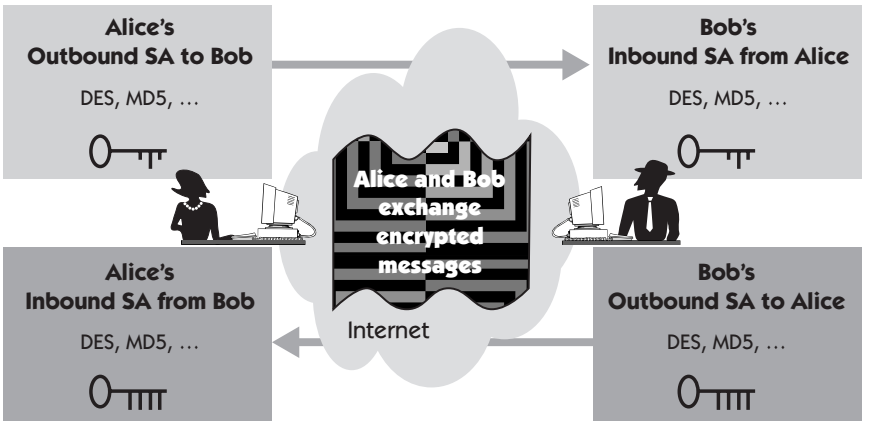

Figure 21-7 Alice and Bob exchange encrypted messages protected by SA-2 parameters.

Some think IPsec's versatility makes it too complex.

IPsec is still evolving, and there are some controversial issues. For example, some experts have criticized IPsec's requirement of inbound and outbound SAs as adding unnecessary complexity. IKE already negotiates SAs in pairs and assures a separate secret key in each direction. Additionally, Alice and Bob should agree to use the most secure encryption method they can for every message they exchange. For example, with respect to the preceding paragraph, if Alice and Bob can use Triple DES, it makes little sense for Alice's outbound SA-2 to use anything weaker (i.e., DES). Critics strongly argue against "needless" complexity, which is often associated with potential security holes.

Other experts admit that it's unfortunate that development of the two parts of IPsec (IKE and bulk data encryption) has not always been coordinated perfectly. Additionally, they counter that although most cryptographic traffic is equally protected in both directions, that's not always the case. For example, a company may distribute secret materials to many regional offices over the Internet in one-way protected traffic. Because the return traffic is not necessarily protected, different inbound and outbound SAs are appropriate.

Protocol and Mode

Definitions: protocol and mode attributes

IPsec bulk encryption provides more options than does SSL/TLS. For example, IPsec offers four confidentiality and message integrity options. The *protocol* attribute controls whether the data packet is protected by confidentiality or message integrity (or both). The *mode* attribute controls how much of the data packet is protected by these assurances.

Protocol:
ESP
AH
Mode:
tunnel
transport

The protocol choices are formally called Encapsulating Security Protection (ESP) and Authentication Header (AH).[8] These IPsec options are most often referred to by their initials rather than by their long formal names. The mode choices are called *tunnel* and *transport*.

Because there are two protocol choices and two mode choices, an IPsec data packet must be protected by one of the four choices shown in Table 21-2.[9] All data packets transmitted under the guise of an SA-2 (bulk encryption parameters) must adhere to the protocol/mode selection.

Table 21-2 IPsec data packets must be protected by one of these four choices.

ESP + Transport	ESP + Tunnel
AH + Transport	AH + Tunnel

As you can see in Figure 21-8, the most robust protection possible with a single SA uses the ESP protocol and tunnel mode. ESP offers both message integrity (authentication) and confidentiality, whereas AH offers only message integrity.[10] Tunnel mode encrypts more of the data packet than does transport mode. Many vendors use ESP in tunnel mode to implement their VPN products.

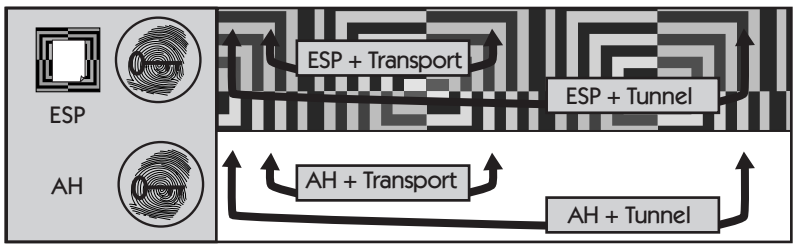

Figure 21-8 IPsec protocol and mode options. ESP offers confidentiality; AH does not. Tunnel mode protects more data than transport mode. ESP in tunnel mode protects the most.

8. IPsec literature uses the term *authentication* to refer to authenticating the origin and integrity of the message sent. Recall from Chapter 7 that integrity is also called "message authentication." In part, cryptographers reason this way: If a message is altered in transit, it means that the altered message came from a new originator.
9. IPsec also permits a "wildcard" option, but it is seldom mentioned in the IETF standards.
10. AH authenticates slightly more of the data packet than ESP.

Virtual Private Networks

Routers or firewalls (we'll refer to them as gateway computers) can use ESP in tunnel mode to hide the addresses of internal computers from the outside world. For example, as shown in Figure 21-9, Alice, a user in her company AliceDotComStocks, works at a computer behind her gateway/firewall computer. Similarly, Bob is a user behind the gateway at HxMel.com. When Alice wishes to establish a secure session with Bob, her gateway automatically negotiates security parameters (SAs) with Bob's gateway. In this scenario, all the IPsec processing is done on their respective gateways; BlackHat knows only that the gateways are communicating and not which particular computers behind the gateways are communicating. This is an example of a simple VPN. The mechanics are discussed in Appendix B.

It's instructive to briefly mention two other competing VPN technologies: Point to Point Tunneling Protocol (PPTP) and Layer 2 Tunneling Protocol (L2TP).

An industry group primarily headed by Microsoft and 3Com created PPTP; it's supported across the Microsoft Windows product line. PPTP supports authentication and confidentiality between a client and a gateway or between two gateways without using public keys. Although early versions of PPTP had significant problems (e.g., poor choice of hash functions), most of the security holes have been patched. But experts feel that PPTP is still vulnerable to an offline password-guessing attack (see http://www.counterpane.com) Microsoft advises PPTP for simple low cost VPNs.

L2TP is a combination of PPTP and another protocol, L2F, created by Cisco Systems. Microsoft advocates using L2TP in concert with IPsec because they feel IPsec doesn't yet have good user authentication standards for client to gateway communications. Note they agree that IPsec is fine for gateway to gateway (machine-to-machine) communications. So L2TP sets up the session and hands it off to IPsec for key negotiation and encryption.

Here's a brief overview of each IPsec protocol and mode, followed by some examples. There's more discussion in Appendix B.

Protocols

ESP is the more robust of the two protocols because it offers both confidentiality and message integrity. Alice and Bob can agree to use ESP for confidentiality and/or authentication, but they must choose at least one assurance.

AH provides only message integrity; it doesn't provide confidentiality. But AH authenticates slightly more of the message than does ESP.

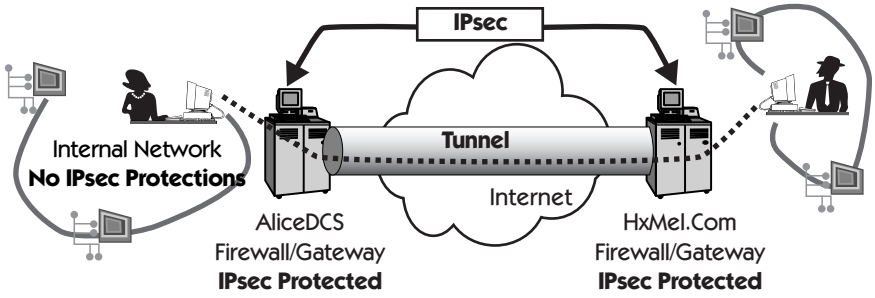

Figure 21-9 Example of a simple virtual private network. IPsec-enabled gateway computers act as protective proxies for Alice and Bob.

Modes

Transport mode protection can be used only between two end host computers; it cannot be used if one of the computers is acting as a gateway that forwards the data packets to their final destination. Transport mode protection uses less bandwidth than tunnel mode because tunnel mode usually appends more data.[11]

Tunnel mode protection can be used in any IPsec-enabled computer, and it must be used when either end SA is a gateway—that is, if either end acts as a proxy for the final destination of the data packet. As shown in Figure 21-9, tunnel mode hides Alice's and Bob's IP addresses from BlackHat.

ESP Examples

Figure 21-10 is an overview of ESP in transport mode and tunnel mode.

ESP in Transport Mode

End to end encryption, authentication

To use transport mode, Alice and Bob must have IPsec installed and must act as host (final destination) computers. Their computers perform encryption/decryption and authentication/verification.

The ESP protocol in transport mode encrypts and authenticates application data (e.g., e-mail) but does not protect the IP addresses. The gateways (AliceDotComStocks and HxMel.com) allow traffic to flow directly from Alice to Bob (and vice versa). Note that because the source address (from

11. Technical note: Tunnel mode adds an IP header which can be used to conceal the ultimate source and destination. See Appendix B.

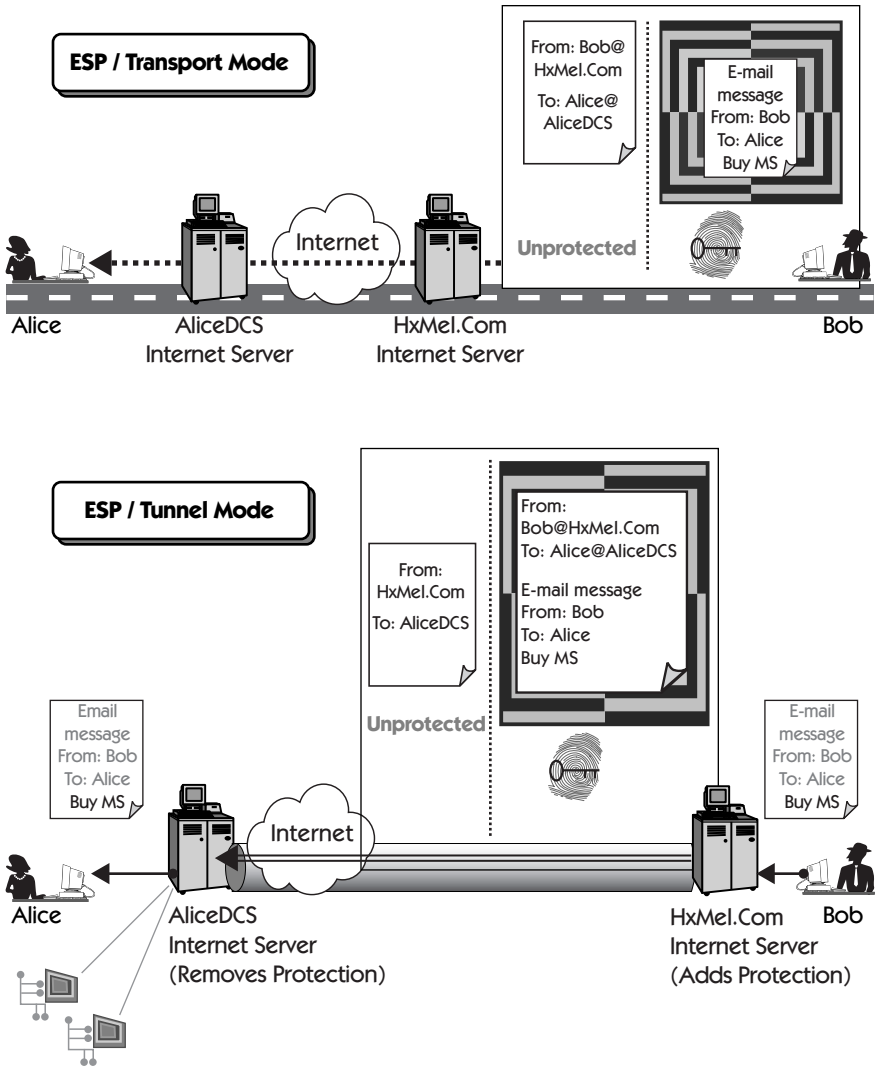

Figure 21-10 ESP in transport mode and tunnel mode.

Bob@HxMel.com) and destination address (to Alice@AliceDotComStocks) are not encrypted, BlackHat can sniff the line (perform traffic analysis) and figure out that Alice and Bob are communicating.

ESP in Tunnel Mode

Firewall to firewall encryption, authentication, and hiding Alice and Bob from BlackHat

IPsec installed on the gateway (firewall) computers acts as a proxy for Alice and Bob. The gateway computers perform encryption/decryption and authentication/verification and then pass the unprotected data packets to Alice (or Bob), who doesn't necessarily need IPsec installed on her (or his) computer.

The ESP protocol in tunnel mode encrypts as much as in transport mode and, in addition, conceals the final source and destination (e.g., Alice's and Bob's Internet addresses). Note that Alice's and Bob's names have been removed from the IP addresses. BlackHat knows only that traffic is flowing from AliceDotComStocks to HxMel.com; he can't figure out which computers behind the gateway are exchanging messages.

AH Examples

Figure 21-11 shows how the AH protocol works in transport mode and tunnel mode.

AH in Transport Mode

End to end authentication

The operation of the AH protocol in transport mode is similar to that of ESP in transport mode but without encryption. Alice and Bob must have IPsec installed and must act as host computers. Their computers perform authentication/verification.

AH authenticates more of the data packet than does ESP. Note that Bob@HxMel.com (and Alice@AliceDotComStocks) is authenticated; ESP does not authenticate (protect) this data.

AH in Tunnel Mode

Firewall to firewall authentication

The operation of AH in tunnel mode is similar to ESP in tunnel mode, but AH does not encrypt and conceal Alice's and Bob's addresses.This means that BlackHat can still see the ultimate source (Bob) and destination (Alice) of the e-mail. Tunnel mode AH offers limited benefit for the increased authentication overhead.

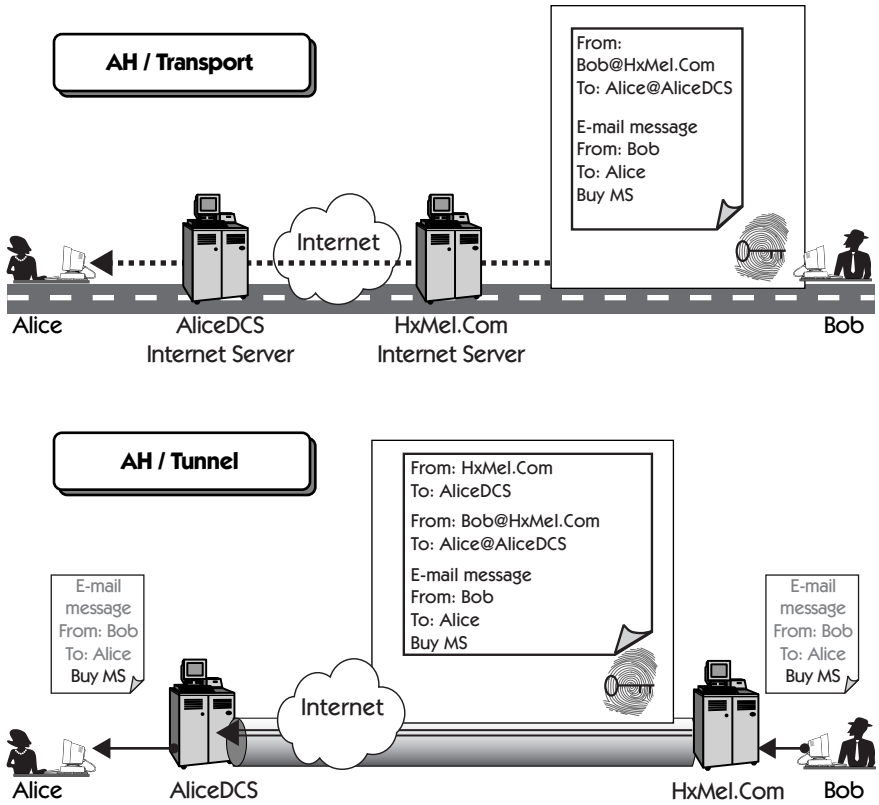

Figure 21-11 AH in transport mode and tunnel mode.

Management Control

Definition: security policy database, policies

Every data packet that leaves or enters an IPsec implementation must comply with the rules found in each IPsec implementation's *security policy database* (SPD). The SPD is the tool IPsec managers use to specify whether and how their computers are allowed to interact with other Internet computers. The SPD specifies rules, called *policies*, that govern the IPsec security provisions between computers. Here's a simplified example.

Definition: selectors

Let's say that Alice (at AliceDotComStocks) wants to send Bob (at HxMel) some data using the file transfer protocol (FTP). From Alice's perspective, the

SA-2 that must be negotiated with Bob must comply with rules in her SPD. The particular SPD policies (rules) that Alice must comply with are selected according to the attributes in Alice's data packets to Bob:

- The source of the data (Alice's computer, identified by her IP address)
- The destination of the data (Bob's computer, identified by his IP address)
- The protocol she's using to send the data (FTP)
- The name of the person to whom the data is being sent (Bob@HxMel.com)
- The source port (Alice's FTP port)
- The destination port (Bob's FTP port)

Each of the six attributes is called a *selector*.

Discard, process, bypass

A particular policy must make one of three choices; Alice's data to Bob is either discarded, subjected to IPsec processing, or bypassed. "Discarded" means that outbound data packets are not allowed to exit (or inbound packets are not allowed to enter). "Subjected to IPsec processing" means that the SPD has identified protection rules with which to process the data packet (e.g., ESP, tunnel mode, Triple DES, and so on). "Bypassed" means that IPsec has determined that the data packet should be allowed to exit (or enter) with no IPsec processing.

In any particular policy, five selectors can be blank, but at least one (usually the source of the data) must be "filled in."

Data flow control

IPsec managers make policies that enumerate required security provisions to and from their particular IPsec-protected network, and thus they control the flow of data. Any data packet that seeks to enter or leave is checked against the SPD and must comply with at least one policy. If no policy is found after the selectors are evaluated, the packet is discarded.

Configuring SPD policies

One of IPsec's strengths is its ability to select and configure multiple security policies for any particular computer or a network of computers. IPsec can configure each IP source address (e.g., Alice), IP destination address (e.g., Bob), protocol (e.g., FTP), and so on. SPD selectors can be broadly or narrowly configured. For example, a broad configuration might require that all traffic leaving Alice's computer use ESP, tunnel, Triple DES, and so on. A narrow configuration might allow all users, except Alice, to communicate with Bob using any protocol, whereas Alice must use FTP.

Implementation Incompatibilities and Complications

SSL/TLS is a stable enough standard that AliceDotComStocks can interact with almost any SSL/TLS standard implementation. This means that Bob can

use Netscape's or Microsoft's SSL/TLS implementation and get a secure cryptographic connection to AliceDotComStocks.

IPsec isn't as widespread as SSL, and there are still vendor compatibility problems. If AliceDotComStocks and HxMel.com have IPsec implementations from different vendors, they may not be able to set up a secure connection. For example, currently IPsec mandates support of only DES; each vendor can include support for additional cipher methods (e.g., Rijndael, Triple DES, etc.) but it's not required. Alice and Bob can communicate only if they can agree on an SA-1 and SA-2.

Security personnel agree that IPsec is complicated. In large part, that's because IPsec delivers a much wider range of cryptographic services with many more options than does SSL/TLS. The IETF, the standards body for both IPsec and SSL/TLS, has about 10 IPsec documents for every SSL/TLS document.

Although cryptographers argue over IPsec's value in its current form, they agree that it is the best protocol for delivering Internet communication security at present. After evaluating IPsec in 1999, *Applied Cryptography* author Bruce Schneier, together with Niels Ferguson, wrote, "We strongly discourage the use of IPsec . . . . However, we even more strongly discourage any current alternatives, and recommend IPsec when the alternative is an insecure network."

Review

IPsec authenticates data entering and encrypts data leaving an IPsec-enabled computer. Its cryptographic protections are delivered to the user as unobtrusively as possible.

The current IPsec standard can be visualized as having two parts. The first part, IKE, manages authentication and key exchange. The second part manages the bulk encryption process.

IKE is a two-phase protocol. The first phase sets up a secure authenticated communication channel; phase 1 establishes encryption parameters that are used to protect the second phase. The second phase makes encryption parameters that are used in IPsec part 2, bulk encryption. Two-phase protocol key management enables quick changes to encryption parameters.

IPsec bulk encryption offers confidentiality and message integrity protections in four potential configurations; two protocols (ESP and AH) and two modes (tunnel and transport). Many vendors of virtual private network products implement their products using IPsec's ESP protocol in tunnel mode.

Management control uses the SPD to make policies. Policies control if and how computers communicate.

Although IPsec has some controversial issues, most of its critics agree that it's currently the best possible solution.