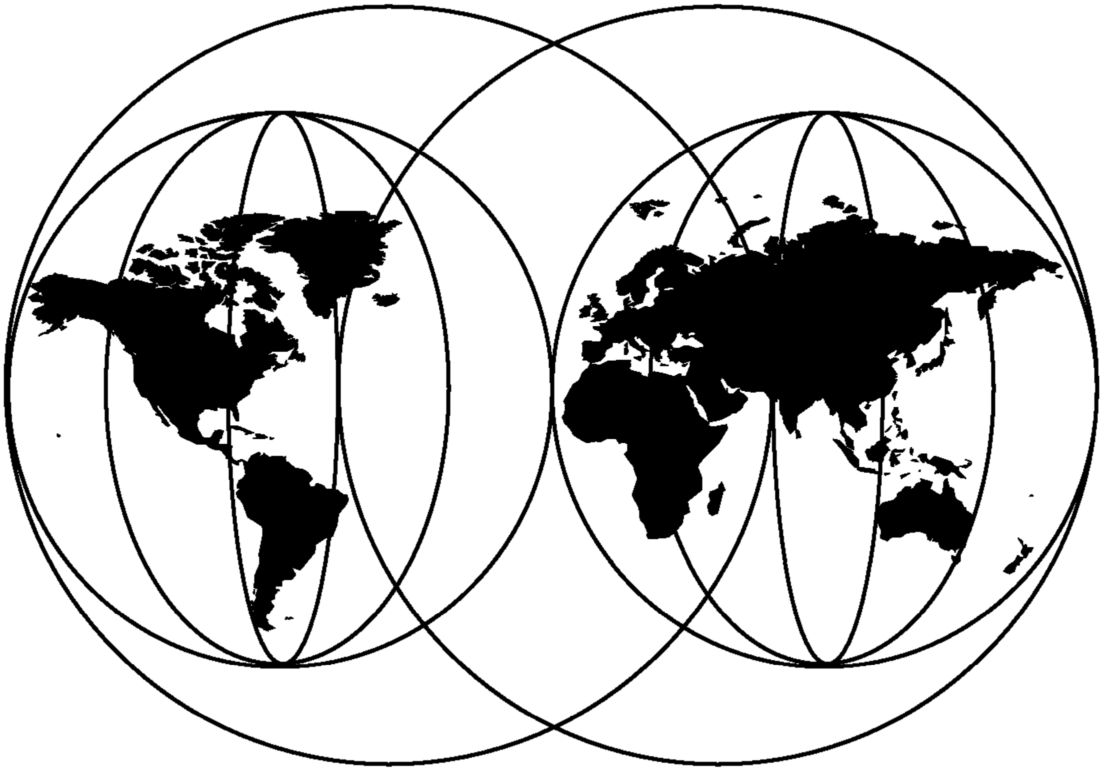
A Comprehensive Guide to Virtual Private Networks, Volume I: IBM Firewall, Server and Client Solutions

Martin Murhammer, Tim Bourne, Tamas Gaidosch,
Charles Kunzinger, Laura Rademacher, Andreas Weinfurter



International Technical Support Organization

http://www.redbooks.ibm.com

This book was printed at 240 dpi (dots per inch). The final production redbook with the RED cover will be printed at 1200 dpi and will provide superior graphics resolution. Please see "How to Get ITSO Redbooks" at the back of this book for ordering instructions.

SG24-5201-00

International Technical Support Organization

SG24-5201-00

A Comprehensive Guide to Virtual Private Networks, Volume I: IBM Firewall, Server and Client Solutions

June 1998

Take Note!

Before using this information and the product it supports, be sure to read the general information in Appendix A, "Special Notices" on page 207.

First Edition (June 1998)

This edition applies to the following IBM software products:

- IBM eNetwork Firewall V3.1 for AIX and IBM eNetwork Firewall V3.2 for AIX
- IBM AIX V4.3
- IBM TCP/IP V4.1 for OS/2
- IBM eNetwork Communications Suite V1.1 for Windows
- IBM OS/390 V2R5

This edition covers the IPSec components of the listed products, as well as IPSec and other VPN features of the IBM 2210 and IBM 2216 Nways Multiprotocol Routers.

Comments may be addressed to:
IBM Corporation, International Technical Support Organization
Dept. HZ8 Building 678
P.O. Box 12195
Research Triangle Park, NC 27709-2195

When you send information to IBM, you grant IBM a non-exclusive right to use or distribute the information in any way it believes appropriate without incurring any obligation to you.

Contents

Figures

Tables

Preface

This redbook describes how to implement virtual private networks (VPNs) based on authentication and encryption as defined in the IP Security Architecture (IPSec) standard and draft documents.

This redbook will help readers to identify the benefits of VPNs in general and the IBM eNetwork VPN solutions in particular, and then to successfully deploy VPNs.

The most commonly used encryption algorithms and handshaking protocols are explained as a general introduction to IP security.

Scenarios describe how to set up IP tunnelling via existing IP networks and the Internet to effectively implement secure and private conversations over public networks. The scenarios are based on the latest available IBM server, client, firewall and router technologies to illustrate point-to-point (client-to-server, server-to-server or client-to-client), branch office (LAN-LAN) and remote user (client-LAN) environments.

An outlook is provided on further development in this area, including certificate and key management frameworks such as Internet Key Exchange (IKE), formerly referred to as ISAKMP/Oakley.

The Team That Wrote This Redbook

This redbook was produced by a team of specialists from around the world working at the Systems Management and Networking ITSO Center, Raleigh. The leader of this project was Martin W. Murhammer.

Martin W. Murhammer is a Senior I/T Availability Professional at the Systems Management and Networking ITSO Center, Raleigh. Before joining ITSO in 1996, he was a Systems Engineer in the Systems Service Center at IBM Austria. He has 13 years of experience in the personal computing environment including such areas as heterogeneous connectivity, server design, system recovery, and Internet solutions. He is a Certified OS/2 Engineer and a Certified LAN Server Engineer and has previously coauthored six redbooks during residencies at the ITSO Raleigh and Austin Centers.

Tim A. Bourne is an Advisory Software Engineer at the IBM PC Company. He has 10 years of experience in software design and development including implementation of Internet protocols, air traffic control and embedded communication systems.

Tamas Gaidosch is an I/T Architect in IBM Hungary. He specializes in networking software and e-business solutions in the banking industry. Tamas has five years of experience in networked computing environments and systems administration. He holds a Master′s degree in Computer Science. His areas of expertise include operating systems (OS/2 LAN Server, Windows NT, AIX), networks (TCP/IP, X.25) and self-service banking software.

Charles Kunzinger is a Senior Engineer in Research Triangle Park, with responsibility for the technical integrity of IBM′s Virtual Private Network line of products. He has worked for IBM since 1967 in various development, advanced technology, and architecture groups. For the last ten years, he has had extensive experience in the development of network layer open standards, and has represented IBM in various open standards bodies, covering such areas as interdomain routing, mobile-IP, and wireless communications, and has contributed to the development of open security standards in each of these fields.

Laura Rademacher is a member of the VPN Brand Management group located in Research Triangle Park, NC, and has spent much of her marketing career on the promotion of VPN technology and solutions. Prior to joining the marketing organization, she was in the TCP/IP technology area, focusing her time mainly on the education and advancement of Internet security. Laura has 15 years of experience with IBM.

Andreas Weinfurter is an Advisory I/T Availability Professional at the IBM System Services Center in Salzburg, Austria. After joining IBM in 1988 he worked as an instructor at the IBM education center in Vienna, where he was responsible for the PC curriculum and held mainly classes on OS/2 and networking. His primary areas of work for the past six years have been AIX and TCP/IP with a strong focus on firewalls during the last two years. Andreas holds a Master′s degree in Computer Science from the Vienna University of Technology.

Thanks to the following people for their invaluable contributions to this project:

Tim Kearby, Karl Wozabal, Jorge Ferrari, Margaret Ticknor,
Linda Robinson, Shawn Walsh, Kathryn Casamento
Systems Management and Networking ITSO Center, Raleigh

Steve Gardner
International Technical Support Organization Center, Austin

Bob Tunstall, Vach Kompella, Jaime Claypool, Steven Lingafelt
Linwood Overby, Cindy Stone-Rutherford
IBM Research Triangle Park

Jackie Wilson, Chris Wenzel, Shay Hoffmaster
IBM Austin

Richard Planutis
IBM Endicott

Kacir Samra
IBM Brazil

Steven Boelaars
IBM Netherlands

Comments Welcome

Your comments are important to us!

We want our redbooks to be as helpful as possible. Please send us your comments about this or other redbooks in one of the following ways:

- Fax the evaluation form found in “ITSO Redbook Evaluation” on page 219 to the fax number shown on the form.
- Use the electronic evaluation form found on the Redbooks Web sites:

For Internet users	`http://www.redbooks.ibm.com/`
For IBM Intranet users	`http://w3.itso.ibm.com/`

- Send us a note at the following address:

  `redbook@us.ibm.com`

Part 1. VPN Description and Background

Chapter 1. Virtual Private Networks (VPN) Overview

The Internet has become a popular, low-cost backbone infrastructure. Its universal reach has led many companies to consider constructing a secure virtual private network (VPN) over the public Internet. The challenge in designing a VPN for today′s global business environment will be to exploit the public Internet backbone for both intra-company and inter-company communication while still providing the security of the traditional private, self-administered corporate network.

In this chapter, we begin by defining a virtual private network (VPN) and explaining the benefits that customers can achieve from its implementation. After discussing the security considerations and planning aspects, we then describe the VPN solutions available in the market today.

1.1 VPN Introduction and Benefits

With the explosive growth of the Internet, companies are beginning to ask: ″How can we best exploit the Internet for our business?″ Initially, companies were using the Internet to promote their company′s image, products, and services by providing World Wide Web access to corporate Web sites. Today, however, the Internet potential is limitless, and the focus has shifted to e-business, using the global reach of the Internet for easy access to key business applications and data that reside in traditional I/T systems. Companies can now securely and cost-effectively extend the reach of their applications and data across the world through the implementation of secure virtual private network (VPN) solutions.

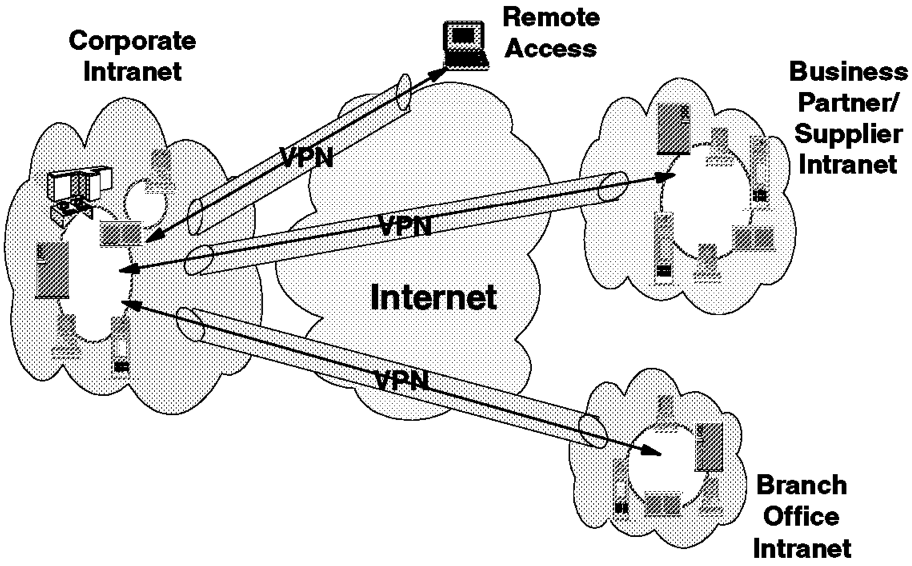


Figure 1. Virtual Private Networks

A virtual private network (VPN) is an extension of an enterprise′s private intranet across a public network such as the Internet, creating a secure private connection, essentially through a private *tunnel*. VPNs securely convey information across the Internet connecting remote users, branch offices, and business partners into an extended corporate network, as shown in Figure 1. Internet Service Providers (ISPs) offer cost-effective access to the Internet (via direct lines or local telephone numbers), enabling companies to eliminate their current, expensive leased lines, long-distance calls, and toll-free telephone numbers.

A 1997 VPN Research Report, by Infonetics Research, Inc., estimates savings from 20% to 47% of wide area network (WAN) costs by replacing leased lines to remote sites with VPNs. And, for remote access VPNs, savings can be 60% to 80% of corporate remote access dial-up costs. Additionally, Internet access is available worldwide where other connectivity alternatives may not be available.

The technology to implement these virtual private networks, however, is just becoming standardized. Some networking vendors today are offering non-standards-based VPN solutions that make it difficult for a company to incorporate all its employees and/or business partners/suppliers into an extended corporate network. However, VPN solutions based on Internet Engineering Task Force (IETF) standards will provide support for the full range of VPN scenarios with more interoperability and expansion capabilities.

The key to maximizing the value of a VPN is the ability for companies to evolve their VPNs as their business needs change and to easily upgrade to future TCP/IP technology. Vendors who support a broad range of hardware and software VPN products provide the flexibility to meet these requirements. VPN solutions today run mainly in the IPv4 environment, but it is important that they have the capability of being upgraded to IPv6 to remain interoperable with your business partner′s and/or supplier′s VPN solutions. Perhaps equally critical is the ability to work with a vendor who understands the issues of deploying a VPN. The implementation of a successful VPN involves more than technology. The vendor′s networking experience plays heavily into this equation.

1.2 Security Considerations for VPNs

The use of VPNs raises several security concerns beyond those that were present in traditional corporate networks. A typical end-to-end data path might contain:

- Several machines not under control of the corporation (for example, the ISP access box in a dial-in segment and the routers within the Internet).
- A security gateway (firewall or router) that is located at the boundary between an internal segment and an external segment.
- An internal segment (intranet) that contains hosts and routers. Some could be malicious, and some will carry a mix of intra-company and inter-company traffic.
- An external segment (Internet) that carries traffic not only from your company′s network but also from other sources.

In this heterogeneous environment, there are many opportunities to eavesdrop, to change a datagram′s contents, to mount denial-of-service attacks, or to alter a datagram′s destination address, as outlined in the following sections. The IBM solutions provide the tools to counter these threats.

Let us have a look at a typical end-to-end path next so that we will be able to understand the security considerations raised with common scenarios.

1.2.1 A Typical End-to-End Path

To understand the issues with VPN end-to-end security, we look at the elements along an end-to-end path. While not all the elements may appear in a given path, some of them will appear in every VPN configuration. End-to-end traffic will usually flow over a mix of three basic segments: a dial-in segment, an external segment (Internet), and an internal segment (intranet).

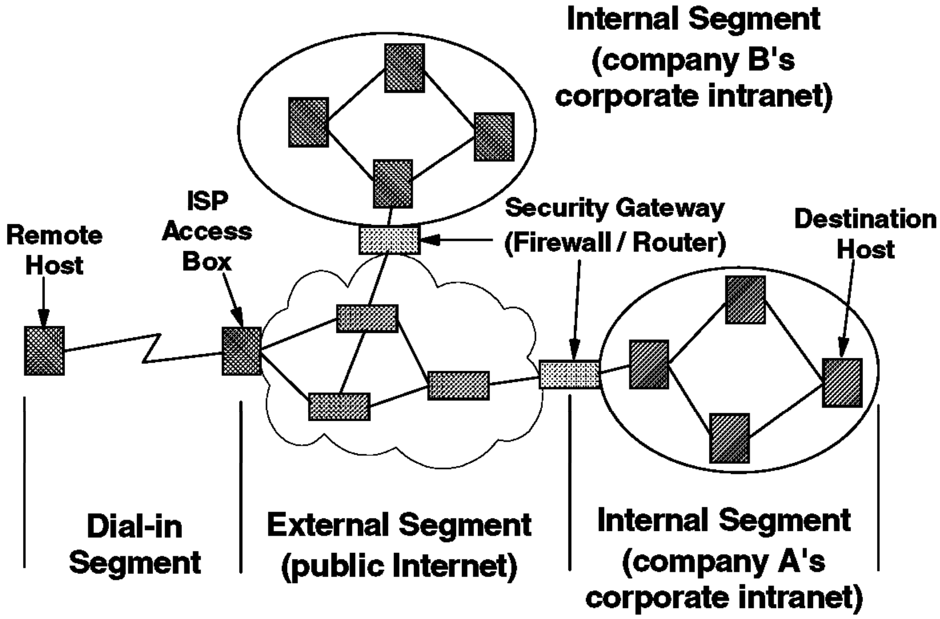

Internal Segment (company B's corporate intranet)
ISP Access Box
Remote Host
Security Gateway (Firewall / Router)
Destination Host
Dial-in Segment
External Segment (public Internet)
Internal Segment (company A's corporate intranet)

Figure 2. Typical Elements in an End-to-End Path

As shown in Figure 2, a path might include a first-hop dial-in connection to an Internet Service Provider (ISP), who in turn uses the backbone public Internet to carry the user′s traffic back to a gateway at the perimeter of the corporate network. Then, the traffic eventually flows within an intranet to its ultimate destination. As we also see in Figure 2, inter-company communication can create a path that includes two separate intranets (for example, company A′s and company B′s).

For discussion purposes in this redbook, we refer to these elements as outlined below:

- **Dial-in Segment:** In today′s environment, remote access has become a necessity. Both work-at-home and on-the-road employees want convenient and secure dial-in access to their company′s networks; and sometimes they even need to communicate with hosts located inside another company′s network. We refer to both work-at-home and on-the-road users as *remote users.* This segment extends from a remote user′s machine to an access box provided by the ISP. The protocols and procedures used on this link are specified by the Internet Service Provider. Today, most ISPs support the Point-to-Point Protocol (PPP) suite of protocols on this segment.
- **External Network (Internet):** The Internet is not owned or operated by any single entity, but is a collection of distinct routing domains, each operated by a different authority. The unifying factor is the standardized IP communications protocols defined by the Internet Engineering Task Force (IETF). The Internet Protocol (IP) suite of protocols will route data traffic at the network layer over a path that may span several ISPs′ routing domains.

Since IP is a connectionless technology, each user datagram could potentially follow a different path. And in fact, traffic from several different companies could all flow simultaneously through a given backbone router in the Internet. For example, a datagram that originated in company A′s intranet and a datagram that originated in company B′s intranet could both flow through a common router located somewhere in the Internet. A company′s traffic on the Internet can no longer be considered to be isolated from the outside world, as it would have been on a dedicated private network, since flows from different VPNs will be intermixed on the Internet backbone.

- **Internal Network (intranet):** This segment appears at an endpoint of the communications path. It is under the control of the corporation, who typically operates and manages it. Traditionally, almost all traffic flowing within a corporate network was generated by the corporation′s employees; very little traffic entered or exited the corporate network; and the protocols in the intranet were proprietary.

  Today, IP is becoming a popular protocol for use within corporate intranets, and data traffic enters and exits the corporate intranet regularly (consider Web browsers, ftp, or telnet applications). In today′s world of e-business, there are emerging requirements for external suppliers and business partners to have access to data stored on another company′s internal servers. Since traffic flowing within an intranet at any given time may have been generated by several different companies, today it may no longer be possible to categorize a given intranet as *trusted* or *untrusted*. A company may consider its own intranets to be trusted, but at the same time its business partners may consider it to be untrusted. In this environment, a VPN designer may need to provide network security functions both on the intranet segments and on the Internet segment.

As shown in Figure 2 on page 5, there are four classes of machines that occur along the path:

- Remote hosts (dial-up)
- Fixed hosts (sources and destinations, or clients and servers)
- ISP access box
- Security gateways (firewalls and/or routers)

Protocols in these machines are used to provide address assignment, tunneling, and IP security. Viable security solutions can be constructed by deploying IP security in some combination of remote hosts, firewalls, routers, and fixed hosts. But since each company should be responsible for its own security, there is no requirement for the ISP boxes or the routers in the Internet backbone to support IP security.

1.2.2 Exposures in a Dial-In Segment

The dial-in segment in Figure 2 on page 5 delivers a user′s data traffic directly to an Internet Service Provider (ISP). If the data is in cleartext (that is, not encrypted), then it is very easy for the ISP to examine sensitive user data, or for an attacker to eavesdrop on the data as it travels over the dial-in link.

Link-layer encryption between the remote host and the ISP can protect against passive eavesdropping, but it does not protect against a malicious ISP. Since the ISP can decrypt the user′s data stream, sensitive data is still available to the ISP in cleartext format.

1.2.3 Exposures in the Internet

In some remote-access scenarios, an ISP builds a tunnel to extend the reach of the PPP connection so that its endpoints will be the access box and the security gateway. If the tunneling protocol does not incorporate robust security features, a malicious ISP could easily build a tunnel that terminates somewhere other than at the correct security gateway (see Figure 3). Thus, user′s data could be delivered via a false tunnel to a malicious impostor gateway where it could be examined or even altered.

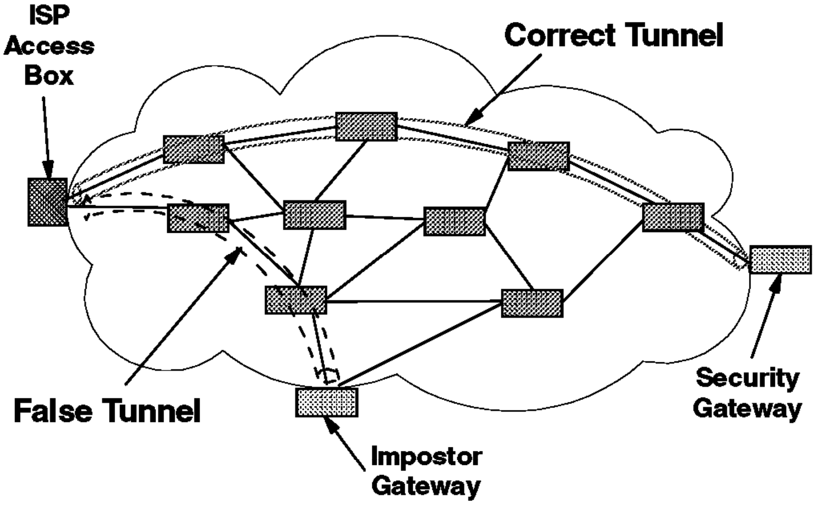


Figure 3. Exposures in the External (Internet) Segment

There are also dangers as the datagram travels within the tunnel. As illustrated in Figure 3, user datagrams pass through routers in the Internet as they travel along a path toward the tunnel endpoint. If the datagrams are in cleartext, any of these routers could easily examine or modify the datagram, and passive attackers could eavesdrop on any of the links along the path.

Link-by-link encryption at each hop in the Internet backbone can thwart eavesdroppers, but does not protect the user′s data from a malicious router, since each router along the path would be capable of decrypting the user′s data stream. Nor does link-by-link encryption protect against false tunnels, since the false tunnel endpoint would have access to cleartext data.

Even popular tunneling protocols such as Layer 2 Tunneling Protocol (L2TP) do not provide robust security. Therefore the IETF has recommended that the tunnel traffic should be protected with the IPSec protocols.

1.2.4 Exposures in a Security Gateway

The security gateway (firewall/router) shown in Figure 2 on page 5 also creates security exposures. Its main purpose is to enforce an access control policy (that is, to accept only the desired inbound traffic, to reject undesired inbound traffic, and to prevent internally generated traffic from indiscriminately leaving the corporate network). The firewall or router is under the control of the corporate network, but an internal attacker still has an opportunity to examine any traffic that the gateway decrypts and then forwards into the intranet in cleartext form.

Non-cryptographic authentication provides some protection against unwanted traffic entering or leaving the network. Common techniques are passwords, packet filtering, and network address translation. However, these can be defeated by a variety of well-known attacks, such as address spoofing, and new attacks are being developed regularly. Each time a new packet filter is designed to thwart a known attack, hackers will devise a new attack, which in turn demands that a new filter rule be generated.

Because the cryptography-based authentication techniques require a long time to break, even with powerful computers, it becomes prohibitively expensive, both in time and in computer power, for a hacker to attempt to attack them. Hence, companies can deploy them with the confidence that they will provide robust protection against a hacker′s attacks.

Link-by-link encryption does not prevent an intermediate box along the path from monitoring, altering, or rerouting valid traffic, since each intermediate box will have access to the cleartext form of all messages. Even host-to-gateway encryption suffers from the same weakness; the gateway still has access to cleartext.

1.2.5 Exposures in an Intranet

Although there is a popular belief that most security threats will occur in the public Internet, there have been studies showing that many of the attacks actually arise internally. Unless every host, gateway, and router within the intranet of Figure 2 on page 5 can be fully trusted, it is possible for a malicious employee to modify an internal box, making it possible to monitor, alter, or reroute datagrams that flow within the corporate network. When data from several different networks flows within the intranet (for example, in the case where the VPN interconnects a manufacturer′s intranet with the intranets of several suppliers) threats within the intranet need to be guarded against. Even if company A trusts that its own intranet is secure, the external supplier or business partner whose traffic must flow through company A′s intranet may not trust it; after all, the partner′s data is at risk if company A′s intranet is in fact compromised in any fashion.

1.2.6 Conclusions

There are security exposures everywhere along an end-to-end path: on the dial-up link, in an ISP′s access box, in the Internet, in the firewall or router, and even in the corporate intranet.

Previously, security solutions were developed to address just a subset of the exposures discussed in this section, but there was no framework that could protect against all these exposures using a single approach.

IP Security Architecture (IPSec) is the first definition of a comprehensive, consistent solution. It can provide end-to-end protection as well as segment-by-segment protection. Based on the work of the Internet Engineering Task Force (IETF) IBM chose to use IPSec for its IBM eNetwork VPN solutions.

The next section gives an overview of some VPN implementations available in the market today. We describe the IPSec components in more detail in Chapter 3, “Description of IPSec” on page 39.

1.3 VPN Solutions in the Marketplace

Vendors′ VPN offerings can be categorized in a number of ways. In our opinion the most important differentiator is the protocol layer on which the VPN is realized. In this context, there are the following different approaches to VPN implementation:

- Network layer-based (IPSec-based)
- Data link layer-based (layer 2-based)

There are other methods that operate on upper layers and complement a VPN solution, such as SOCKS, Secure Sockets Layer (SSL), or Secure Multipurpose Internet Mail Extension (S-MIME). Some vendors′ solutions use only the upper layer protocols to construct a VPN, usually a combination of SOCKS V5 and SSL. In Figure 4 the TCP/IP layered protocol stack is shown, with the VPN related protocols associated to each layer. A description of these can be found in 1.3.3, "Non-IPSec Network Layer-Based Components of a VPN Solution" on page 14.

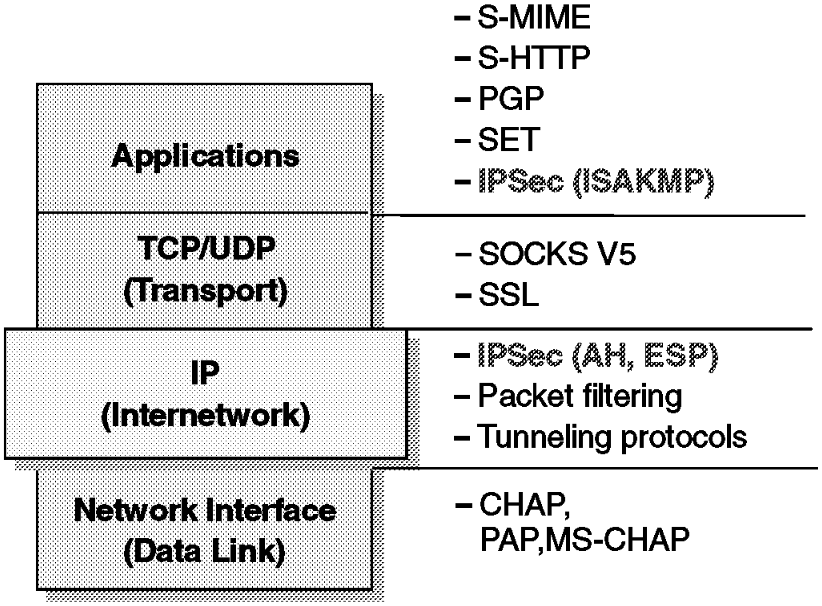


Figure 4. The TCP/IP Protocol Stack and the VPN-Related Protocols

1.3.1 IPSec-Based VPN Solutions

Within the layered communications protocol stack model, the network layer (IP in the case of the TCP/IP stack) is the lowest layer that can provide end-to-end security. Network-layer security protocols provide blanket protection for all upper-layer application data carried in the payload of an IP datagram, without requiring a user to modify the applications.

The solutions are based on the IP Security Architecture (IPSec) open framework, defined by the IPSec Working Group of the IETF. IPSec is called a framework because it provides a stable, long lasting base for providing network layer security. It can accommodate today′s cryptographic algorithms, and can also accommodate newer, more powerful algorithms as they become available. IPv6 implementations are required to support IPSec, and IPv4 implementations are strongly recommended to do so. In addition to providing the base security functions for the Internet, IPSec furnishes flexible building blocks from which robust, secure virtual private networks can be constructed.

The IPSec Working Group has concentrated on defining protocols to address several major areas:

- *Data origin authentication* verifies that each datagram was originated by the claimed sender.
- *Data integrity* verifies that the contents of the datagram were not changed in transit, either deliberately or due to random errors.
- *Data confidentiality* conceals the cleartext of a message, typically by using encryption.
- *Replay protection* assures that an attacker can not intercept a datagram and play it back at some later time without being detected.
- *Automated management of cryptographic keys and security associations* assures that a company′s VPN policy can be conveniently and accurately implemented throughout the extended network with little or no manual configuration. These functions make it possible for a VPN′s size to be scaled to whatever size a business requires.

Note: The above mentioned areas (among others) are the subject of the discipline of cryptography. For a short introduction to cryptography see Chapter 2, "A Short Introduction to Cryptography" on page 23.

The principal IPSec protocols are:

- IP Authentication Header (AH) provides data origin authentication, data integrity, and replay protection.
- IP Encapsulating Security Payload (ESP) provides data confidentiality, data origin authentication, data integrity, and replay protection.
- Internet Security Association and Key Management Protocol (ISAKMP) provides a method for automatically setting up security associations and managing their cryptographic keys.

1.3.1.1 Authentication Header (AH)

The IP Authentication Header provides connectionless (that is, per-packet) integrity and data origin authentication for IP datagrams, and also offers protection against replay. Data integrity is assured by the checksum generated by a message authentication code (for example, MD5); data origin authentication is assured by including a secret shared key in the data to be authenticated; and replay protection is provided by use of a sequence number field within the AH header. In the IPSec vocabulary, these three distinct functions are lumped together and simply referred to by the name *authentication*.

1.3.1.2 Encapsulating Security Payload (ESP)

The IP Encapsulating Security Payload provides data confidentiality (encryption), connectionless (that is per-packet) integrity, data origin authentication, and protection against replay. ESP always provides data confidentiality, and can also optionally provide data origin authentication, data integrity checking, and replay protection. Comparing ESP to AH, one sees that only ESP provides encryption, while either can provide authentication, integrity checking, and replay protection.

When ESP is used to provide authentication functions, it uses the same algorithms used by the AH protocol. However, the coverage is different.

1.3.1.3 Combining the Protocols

Either ESP or AH may be applied alone, in combination with the other, or even nested within another instance of itself. With these combinations, authentication and/or encryption can be provided between a pair of communicating hosts, between a pair of communicating firewalls, or between a host and a firewall.

1.3.1.4 ISAKMP/Oakley

A security association (SA) contains all the relevant information that communicating systems need in order to execute the IPSec protocols, such as AH or ESP. For example, a security association will identify the cryptographic algorithm to be used, the keying information, the identities of the participating parties, etc. ISAKMP defines a standardized framework to support negotiation of security associations (SA), initial generation of all cryptographic keys, and subsequent refresh of these keys. Oakley is the mandatory key management protocol that is required to be used within the ISAKMP framework. ISAKMP supports automated negotiation of security associations, and automated generation and refresh of cryptographic keys. The ability to perform these functions with little or no manual configuration of machines will be a critical element as a VPN grows in size.

Secure exchange of keys is the most critical factor in establishing a secure communications environment—no matter how strong your authentication and encryption are, they are worthless if your key is compromised. Since the ISAKMP procedures deal with initializing the keys, they must be capable of running over links *where no security can be assumed to exist*. That is, they are used to *bootstrap* the IPSec protocols. Hence, the ISAKMP protocols use the most complex and processor-intensive operations in the IPSec protocol suite.

ISAKMP requires that all information exchanges must be both encrypted and authenticated. No one can eavesdrop on the keying material, and the keying material will be exchanged only among authenticated parties.

1.3.1.5 The Vendors

An IPSec-based VPN can be built in many different ways according to the user′s needs. In the general case, a combination of clients, servers, firewalls and routers are using IPSec technology. These components might come from different vendors, thus interoperability is a major requirement.

Without attempting to be complete, here is an enumeration of the active players in this field: Ascend, Bay Networks, Checkpoint, Cisco, Hewlett Packard, IBM, Intel, Sun, 3Com. These companies offer one or more of the following: IPSec-enabled stacks for different operating system platforms, IPSec-enabled firewall software and IPSec-enabled routers. Microsoft has announced IPSec support for Version 5 of their Windows NT operating system.

In the view of IBM, since the needs of companies differ significantly, any VPN implementation is likely to be custom-made, and should therefore include service and support. In order to meet these customer requirements, IBM has developed the eNetwork Virtual Private Network, an industrial strength VPN solution that incorporates a wide range of client, server, firewall and router offerings, with installation and maintenance for ease of use.

Please refer to Chapter 3, “Description of IPSec” on page 39 for a technical description of the IPSec framework and to Chapter 4, “IBM eNetwork VPN Solutions” on page 59 for details on the IBM eNetwork VPN offerings.

1.3.2 Layer 2-Based VPN Solutions

A remote access dial-up solution for mobile users is a very simple form of a virtual private network, typically used to support dial-in access to a corporate network whose users are all company employees. To eliminate the long-distance charges that would occur if a remote user were to dial-in directly to a gateway on the home network, the IETF developed a tunneling protocol, Layer 2 Tunnel Protocol (L2TP). This protocol extends the span of a PPP connection: instead of beginning at the remote host and ending at a local ISP′s point of presence (PoP), the *virtual PPP* link now extends from the remote host all the way back to the corporate gateway. In effect, the remote host appears to be on the same subnet as the corporate gateway.

Since the host and the gateway share the same PPP connection, they can take advantage of PPP′s ability to transport protocols other than just IP. For example, L2TP tunnels can be used to support remote LAN access as well as remote IP access. Figure 5 outlines a basic L2TP configuration:

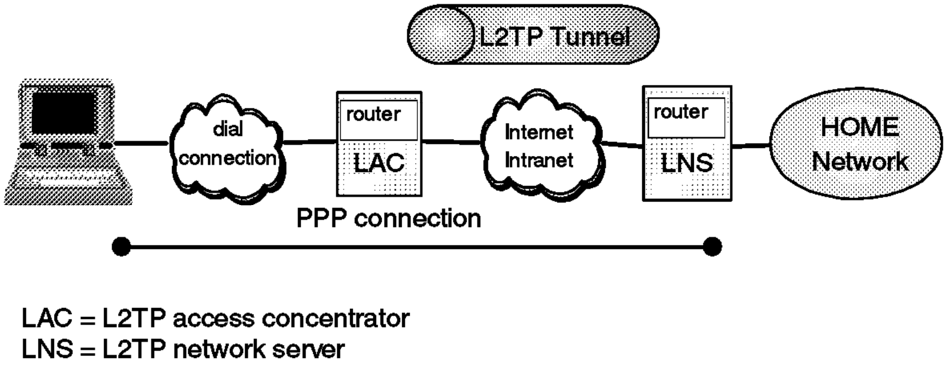


Figure 5. Layer 2 Tunnel Protocol (L2TP) Scenario

Although L2TP provides cost-effective access, multiprotocol transport, and remote LAN access, it does not provide cryptographically robust security features. For example:

- Authentication is provided only for the identity of tunnel endpoints , but not for each individual packet that flows inside the tunnel. This can expose the tunnel to man-in-the-middle and spoofing attacks.
- Without per-packet integrity, it is possible to mount denial-of-service attacks by generating bogus control messages that can terminate either the L2TP tunnel or the underlying PPP connection.
- L2TP itself provides no facility to encrypt user data traffic. This can lead to embarrassing exposures when data confidentiality is an issue.
- While the payload of the PPP packets can be encrypted, the PPP protocol suite does not provide mechanisms for automatic key generation or for automatic key refresh. This can lead to someone listening in on the wire to finally break that key and gain access to the data being transmitted.

Realizing these shortcomings, the PPP Extensions Working Group of the IETF considered how to remedy these shortfalls. Some members proposed to develop new IPSec-like protocols for use with PPP and L2TP. But since this work would have substantially duplicated the more mature work of the IPSec Working Group, the IETF took the position instead to support the use of the existing IPSec protocols to protect the data that flows through an L2TP tunnel.

L2TP is actually another variation of an IP encapsulation protocol. As shown in Figure 6 on page 13, an L2TP tunnel is created by encapsulating an L2TP frame inside a UDP packet, which in turn is encapsulated inside an IP packet whose source and destination addresses define the tunnel′s endpoints. Since the outer encapsulating protocol is IP, clearly IPSec protocols can be applied to this composite IP packet, thus protecting the data that flows within the L2TP tunnel. AH, ESP, and ISAKMP/Oakley protocols can all be applied in a straightforward way.

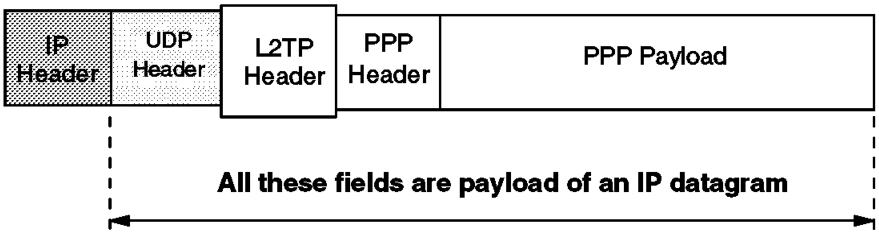


Figure 6. L2TP Tunnel Encapsulation In IP

The following reference provides additional information on how to use IPSec in conjunction with L2TP:

```
http://www.ietf.org/internet-drafts/draft-ietf-pppext-l2tp-security-02.txt
```

In summary, layer 2 tunnel protocols are an excellent way of providing cost-effective remote access. And when used in conjunction with IPSec, they are an excellent technique for providing secure remote access. However, without complementary use of IPSec, an L2TP tunnel alone does not furnish adequate security for the solutions that we discuss later in this redbook.

L2TP is a consensus standard that came from the merging of two earlier tunneling protocols: Point-to-Point Tunneling Protocol (PPTP) and Layer 2 Forwarding (L2F; described in RFC 2341). These earlier protocols did not provide as complete a solution as the L2TP protocol; one addresses tunnels created by ISPs and the other addresses tunnels created by remote hosts. L2TP supports both host-created and ISP-created tunnels. As far as vendors are considered, Microsoft incorporates its proprietary PPTP protocol into its Windows NT and Windows 95 operating systems. Cisco offers L2F and L2TP capabilities in its midrange and high-end router product line. The following IBM products will provide L2TP support (besides IPSec) by June 1998:

- IBM Nways Multiprotocol Routing Services Version 3.1, which is the licensed software for the IBM 2210 router family (L2TP support was in fact already available with Version 2.2)
- IBM Nways Multiprotocol Access Services Version 3.1, which is the licensed software for the IBM 2216 router family

Note: L2TP will not available on the 1x4 models of the IBM 2210 family.

Please see the redbook *A Comprehensive Guide to Virtual Private Networks, Volume II: IBM Nways Router Solutions*, SG24-5234, to be published later this year, for more information on how to implement L2TP scenarios with the IBM 2210 and 2216 routers.

1.3.3 Non-IPSec Network Layer-Based Components of a VPN Solution

The IPSec framework provides security for the network layer of the protocol stack. That is, it provides security functions between pairs of machines that have verifiable network layer identities (such as the IP address, fully qualified domain names, and so on). However, IPSec does not work in isolation. Other protocols and functions can be employed to complement IPSec′s security functions, for example by providing finer granularity over the material to be protected. For example, efficient Certificate Management functions can make IPSec easier to deploy, and upper layer security functions, such as Secure Sockets Layer (SSL) can provide application level security in addition to IPSec′s network layer security. Or a centralized VPN policy directory that can be accessed with a protocol such as Lightweight Directory Access Protocol (LDAP) can make it easier to configure systems correctly without tedious manual operations.

The following sections explain several services and protocols as they relate to IPSec. Designers of VPNs should keep in mind that these techniques are complementary to IPSec, and in many cases can be used in conjunction with IPSec to provide very fine-grained protection of applications in cases where it is needed.

We already took a look at IPSec, which is located at the network layer. Let us see what other functionality is offered at this layer.

1.3.3.1 Network Address Translation

Sometimes globally unique IP addresses are a scarce resource. For example, in Europe, it is especially hard to obtain a globally unique IPv4 address. Other times, a company simply wishes to keep secret the IP addresses of the machines in its intranet (an unlisted address, similar in concept to an unlisted phone number). Both of these situations can be addressed with a function called Network Address Translation (NAT).

Network Address Translation (NAT) is usually implemented in a machine that resides at the boundary of a company′s intranet, at the point where there is a link to the public Internet. In most cases this machine will be a firewall or router. NAT sets up and maintains a mapping between internal IP addresses and external public (globally unique) IP addresses. Because the internal addresses are not advertised outside of the intranet, NAT can be used when they are private (globally ambiguous) addresses, or when they are public (globally unique) addresses that a company wishes to keep secret.

The weakness of NAT in context to VPNs is that by definition the NAT-enabled machine will change some or all of the address information in an IP packet. When end-to-end IPSec authentication is used, a packet whose address has been changed will always fail its integrity check under the AH protocol, since any change to any bit in the datagram will invalidate the integrity check value that was generated by the source.

Within the IETF, there is a working group that is looking at the deployment issues surrounding NAT. This group has been advised by the Internet Engineering Steering Group (IESG) that the IETF will not endorse any deployment of NAT that would lead to weaker security that can be obtained when NAT is not used. Since NAT makes it impossible to authenticate a packet using IPSec′s AH protocol, NAT should be considered as a temporary measure at best, but should not be pursued as a long term solution to the addressing problem when dealing with secure VPNs.

IPSec protocols offer some solutions to the addressing issues that were previously handled with NAT. We see in later scenarios that there is no need for NAT when all the hosts that comprise a given virtual private network use globally unique (public) IP addresses. Address hiding can be achieved by IPSec′s tunnel mode. If a company uses private addresses within its intranet, IPSec′s tunnel mode can keep them from ever appearing in cleartext form in the public Internet, which eliminates the need for NAT.

Note: Be careful about NAT issues in conjunction with VPNs. If you are using private IP addresses and have a need to access public resources on the Internet, you are likely to have a need for NAT. If you are going to deploy an IPSec-based VPN, there are scenarios where using NAT would be detrimental to what you are trying to achieve.

1.3.3.2 Packet Filtering

Packet filtering is a technique that is commonly provided in many firewall and router products and in many routers. Packet filtering relies on having access to cleartext; that is, the contents of the IP datagram can not be encrypted or compressed. The machine examines the contents of an IP packet (typically the IP header and the TCP header, and sometimes even the contents of the TCP payload) looking for things such as source addresses, destination addresses, protocol IDs, port numbers, etc. The firewall or router then applies a set of detailed filtering rules to this information to make a decision on whether to accept or reject the packet.

There are various degrees of complexity in filtering. *Stateless* inspection makes a decision on each packet individually, while *stateful* inspection makes a decision for a given packet based on both the packet itself and its history. For example, the history for the TCP protocol could involve monitoring whether or not the TCP handshake messages occur in the correct order within an acceptable time interval.

The advantage of packet filtering is that it provides excellent granularity for making access control decisions. But this is also one of its weaknesses, since the granularity can only be achieved through the specification of elaborate, detailed filtering rules. Rules development tends to be a tedious, error-prone process. And, even if a robust set of rules is in place, they are vulnerable to relatively crude *spoofing attacks*. As new attacks are discovered, firewall administrators end up on a treadmill. Each attack must be countered with very specific new rules, but these new rules don′t offer protection against the next new attack.

The major drawback to packet filtering techniques in context to VPNs is that they require access to cleartext, both in packet headers (for stateless inspection) and in the packet payloads (for stateful inspection). When encryption is applied, some or all of the information needed by the packet filters may no longer be available. For example:

- In transport mode, ESP will encrypt the payload of the IP datagram, thus precluding the use of stateful inspection techniques.
- In tunnel mode, ESP will encrypt the entire original datagram, both header and payload, thus precluding stateless or stateful inspection of the original datagram.

IPSec offers a way out of this dilemma. Its AH protocol offers a cryptographically robust and spoof-proof way to enforce access control, and its HMAC algorithms are robust enough that they can not be broken by most hackers. The processor power and the time needed to break them are both prohibitively expensive.

In most IPSec-based VPNs, packet filtering will no longer be the principal method for enforcing access control. IPSec′s AH protocol, which is cryptographically robust, will fill that role. Both the number and the complexity of filtering rules will be greatly reduced, and they will be used for fine-tuning only after a packet has already been successfully authenticated by IPSec. And since IPSec′s authentication and encryption protocols can be applied simultaneously to a given packet, strong access control can be enforced even when the data itself is encrypted.

If the security gateway (firewall or router) is the endpoint of the tunnel, it still enables you to use packet filtering between itself and the destination host in the secure network, because the packet filters are evaluated before a packet is sent to the IPSec kernel, that is, before applying authentication and/or encryption.

Modern routers also offer packet filtering on a physical port level which allows you to restrict access to secure networks all together, or to redirect traffic, tunneled or otherwise, to a specific port based on the destination address. In that case, access to cleartext is not required but the practical use in a VPN environment may be limited to a subset of possible configurations.

1.3.3.3 Quality-of-Service (QOS)

In a virtual private network, just as in a conventional network, there will be a desire to provide distinct transport characteristics (quality of service) for packets as they travel from source to destination. The IP protocol provides Type of Service (TOS) bits that can be used for this purpose. The details of how to use these bits is a relatively new item of work in the IETF, so no firm standard solutions exist today. However, looking forward to future requirements, the IPSec′s AH protocol treats the TOS bits as *mutable*, thus allowing them to be changed as needed while an IPSec protected datagram travels throught the Internet. Thus, IPSec is already positioned to take advantage of the emerging QOS work as soon as it matures.

1.3.4 Non-IPSec Application Layer-Based Components of a VPN Solution

Many firewalls provide application gateways. This technique requires the firewall to be aware of those applications that it will permit to flow across the boundary of a corporate intranet. The user connects to the firewall, which terminates the application. Then, the firewall launches another copy of the same application, running it between itself and the external destination. The firewall then provides synchronization between the internal application (user-to-firewall) and the external application (firewall-to-destination).

1.3.4.1 SOCKS

SOCKS is located at the session layer of the OSI model. The client usually connects to the firewall at port 1080. The firewall then establishes a separate session to the destination host, making the client invisible to the destination host. A drawback is that the client applications need to be *socksified*, which means they have to implement the socks protocol, since it is located at a higher layer than IPSEC the performance of socks is slower than IPSEC. On the other hand this higher layer gives it also more possibilities to control the session.

1.3.4.2 Secure Socket Layer (SSL)

Secure Socket Layer (SSL) is an upper-layer mechanism commonly used by Web browser clients and servers to provide peer authentication and encryption of application data. SSL mandates that the server authenticate itself to the client via a certificate-based technique. Authentication of the client to the server is optional in SSL Version 3, but is not commonly used in practice. SSL involves a handshake phase, where certificates are exchanged, session keys are generated, and encryption algorithms are agreed to. After the handshake phase, user data will be exchanged securely without the need for the application to be explicitly modified, other than to invoke the SSL services before actual data transfer begins.

SSL is an end-to-end protocol, and therefore will be implemented in the machines at the endpoints of a given path (typically the client and the server), but it is not implemented in the intermediate machines along a given path (such as routers or firewalls). Although in theory SSL could be used to protect any TCP/IP application, it is almost exclusively used for HTTP. The client uses any non-privileged port and the server uses port 443.

1.3.4.3 Secure HTTP (S-HTTP)

S-HTTP is a security addition to HTTP. It provides authentication and optionally also encryption. Although it is more flexible than SSL, S-HTTP is rarely used in practice as SSL is easier to administer and has proved functionally adequate for most secure Web applications. Web pages that use S-HTTP have a URL starting with https://. The client uses any non-privileged port and the server uses port 80 (such as HTTP).

1.3.4.4 Secure Mail (S-MIME)

Secure Multipurpose Internet Mail Extension (S-MIME) can be thought of as a very specific SSL-like protocol. S-MIME is an application-level security construct, but its use is limited to protecting e-mail via encryption and digital signatures. It relies on public key technology, and uses X.509 certificates to establish the identities of the communicating parties. S-MIME may be implemented in the communicating end systems; it is not used by intermediate routers or firewalls.

1.3.5 Conclusions

Neither network layer-based nor application layer-based security techniques are the best choice for all situations. There will be trade-offs. Network layer security protects the information created by upper layer protocols, but it requires that IPSec be implemented in the communications stack. With network layer security, there is no need to modify existing upper layer applications. On the other hand, if security features are already imbedded within a given application, then the data for that specific application will be protected while it is in transit, even in the absence of network layer security. Therefore security functions must be imbedded on a per-application basis.

There are still other considerations:

- Network layer security gives ″blanket protection″, but this may not be as fine-grained as would be desired for a given application. It protects all traffic and is transparent to users and applications.
- Network layer security does not provide protection once the datagram has arrived at its destination host. That is, it is vulnerable to attack within the upper layers of the protocol stack at the destination machine.

- Application layer security can protect the information that has been generated within the upper layers of the stack, but it offers no protection against several common network layer attacks while the datagram is in transit. For example, a datagram in transit would be vulnerable to spoofing attacks against its source or destination address.
- Application layer security is more intelligent (as it knows the application) but also more complex and slower.

Many cases can occur, each of which needs to be examined on its own merit. It may be desirable to employ a mix of both network layer security techniques and application layer techniques to achieve the desired overall level of protection. For example, you could use an upper layer mechanism such as Secure Sockets Layer (SSL) to encrypt upper layer data. SSL could then be supplemented with IPSec′s AH protocol at the network layer to provide per-packet data origin authentication and protection against spoofing attacks.

1.4 VPN Customer Scenarios

In this section we look at the three most likely business scenarios well suited to the implementation of a VPN solution.

1. Branch office connection network

2. Business partner/supplier network

3. Remote access network

This section provides a general, overview-type description of those scenarios. Technical issues and configuration details are provided in Chapter 5, “Branch Office Connection Scenario” on page 81, Chapter 6, “Business Partner/Supplier Network Scenario” on page 111, and Chapter 7, “Remote Access Scenario” on page 133, respectively.

1.4.1 Branch Office Connection Network

The branch office scenario securely connects two trusted intranets within your organization. This is a key difference, since your security focus is on both protecting your company′s intranet against external intruders and securing your company′s data while it flows over the public Internet. This differs from the business partner/supplier network discussed in 1.4.2, “Business Partner/Supplier Network” on page 19, where the focus is on enabling your business partners/suppliers access to data in your corporate intranet.

For example, suppose corporate headquarters wants to minimize the costs incurred from communicating to and among its own branches. Today, the company may use frame relay and/or leased lines, but wants to explore other options for transmitting their internal confidential data that will be less expensive, more secure, and globally accessible. By exploiting the Internet, branch office connection VPNs can easily be established to meet the company′s needs.

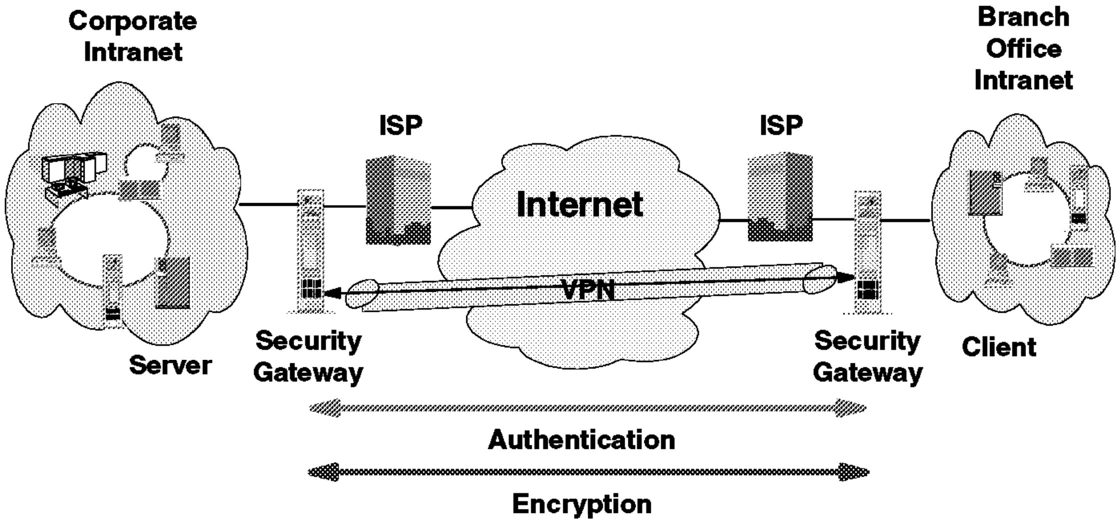


Figure 7. Branch Office Connection Network

As shown in Figure 7, one way to implement this VPN connection between the corporate headquarters and one of its branch offices is for the company to purchase Internet access from an ISP, such as IBM Global Services. IBM eNetwork firewalls, or routers with integrated firewall functionality, or in some cases an IBM server with IPSec capability, would be placed at the boundary of each of the intranets to protect the corporate traffic from Internet hackers. With this scenario, the clients and servers need not support IPSec technology, since the IPSec-enabled firewalls (or routers) would be providing the necessary data packet authentication and encryption. With this approach, any confidential information would be hidden from untrusted Internet users, with the firewall denying access to potential attackers.

With the establishment of branch office connection VPNs, the company′s corporate headquarters will be able to communicate securely and cost-effectively to its branches, whether located locally or far away. Through VPN technology, each branch can also extend the reach of its existing intranet to incorporate the other branch intranets, building an extended, enterprise-wide corporate network.

And, as in the business partner/supplier network scenario, this company can easily expand this newly created environment to include its business partners, suppliers, and remote users, through the use of open IPSec technology.

1.4.2 Business Partner/Supplier Network

Industry-leading companies will be those that can communicate inexpensively and securely to their business partners, subsidiaries, and vendors. Many companies have chosen to implement frame relay and/or purchase leased lines to achieve this interaction. But this is often expensive, and geographic reach may be limited. VPN technology offers an alternative for companies to build a private and cost-effective extended corporate network with worldwide coverage, exploiting the Internet or other public network.

Suppose you are a major parts supplier to a manufacturer. Since it is critical that you have the specific parts and quantities at the exact time required by the manufacturing firm, you always need to be aware of the manufacturer′s inventory status and production schedules. Perhaps you are handling this interaction manually today, and have found it to be time consuming, expensive and maybe even inaccurate. You′d like to find an easier, faster, and more effective way of communicating. However, given the confidentiality and time-sensitive nature of this information, the manufacturer does not want to publish this data on their corporate Web page or distribute this information monthly via an external report.

To solve these problems, the parts supplier and manufacturer can implement a VPN, as shown in Figure 8. A VPN can be built between a client workstation, in the parts supplier′s intranet, directly to the server residing in the manufacturer′s intranet. The clients can authenticate themselves either to the firewall or router protecting the manufacturer′s intranet, directly to the manufacturer′s server (validating that they are who they say they are), or to both, depending on your security policy. Then a tunnel could be established, encrypting all data packets from the client, through the Internet, to the required server.

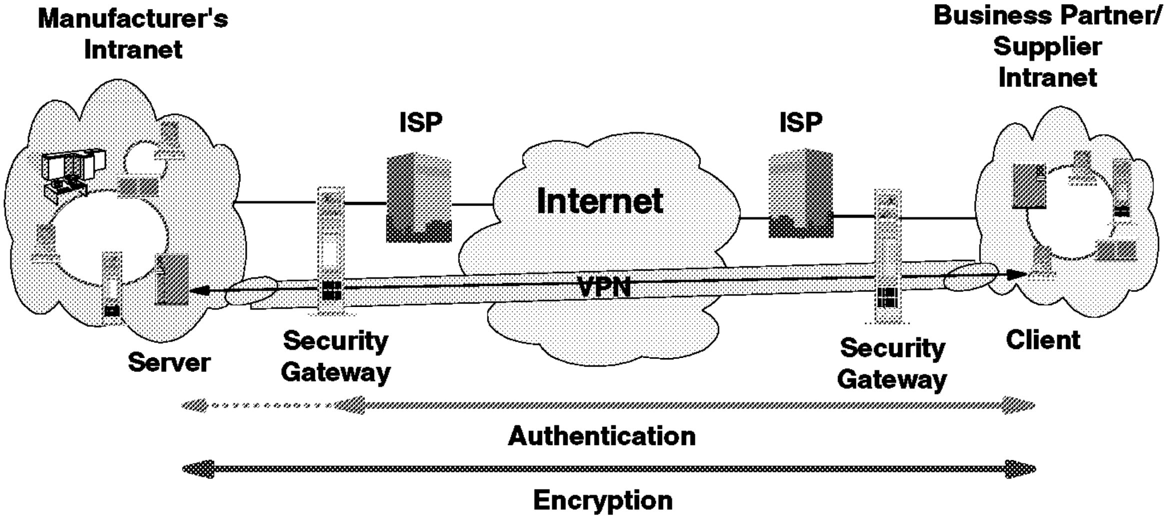


Figure 8. Business Partner/Supplier Network

With the establishment of this VPN, the parts supplier can have global, online access to the manufacturer′s inventory plans and production schedule at all times during the day or night, minimizing manual errors and eliminating the need for additional resources for this communication. In addition, the manufacturer can be assured that the data is securely and readily available to only the intended parts supplier(s).

One way to implement this scenario is for the companies to purchase Internet access from an Internet service provider (ISP), such as IBM Global Services. Then, given the lack of security of the Internet, either an IBM eNetwork firewall or IPSec-enabled router, or an IBM server with IPSec capability can be deployed as required to protect the intranets from intruders. If end-to-end protection is desired, then both the client and server machines need to be IPSec-enabled as well.

Through the implementation of this VPN technology, the manufacturer would easily be able to extend the reach of their existing corporate intranet to include one or more parts suppliers (essentially building an extended corporate network) while enjoying the cost-effective benefits of using the Internet as their backbone. And, with the flexibility of open IPSec technology, the ability for this manufacturer to incorporate more external suppliers is limitless.

Yet, inherent in network expansion are concerns of manageability. Tools should be implemented to ensure your network remains easy to maintain. Management functions to be included in future eNetwork VPN solutions are:

- Policy management
- Automated ISAKMP/Oakley key management capabilities
- Certificate management
- Secure domain name server (DNS)
- Lightweight Directory Access Protocol (LDAP) support

When implementing a VPN, a set of security configuration criteria must be established. Decisions such as which security algorithms are to be used by each IPSec-enabled box and when the keys are to be refreshed are all aspects of policy management. And, with respect to key technology, almost all of today′s currently popular security protocols begin by using public key cryptography. Each user is assigned a unique public key. Certificates, in the form of digital signatures, validate the authenticity of your identity and your encryption key. These certificates can be stored in a public key database, such as a secure DNS, that can be accessible via a simple protocol, such as LDAP.

An automated IP address management system is especially important for VPNs in order to assign and manage your network′s IP addresses. IBM is working with an IP address management company to offer highly centralized control of all network devices in your entire extended intranet. Also, along the lines of managing your IP addresses, network address translation (NAT), available today in the eNetwork Firewall for AIX, allows you to use a globally unique (public) address on the Internet, while enabling you to use private IP addresses within your intranet.

IBM will be incorporating all of these VPN management tools into its eNetwork VPN solutions, which can easily be implemented to meet the needs of your existing and future networking environment. The future of VPN is discussed in Chapter 10, “The Internet Key Exchange (IKE) Protocol” on page 193.

1.4.3 Remote Access Network

A remote user, whether at home or on the road, wants to be able to communicate securely and cost-effectively back to his/her corporate intranet. Although many still use expensive long-distance and toll-free telephone numbers, this cost can be greatly minimized by exploiting the Internet. For example, you are at home or on the road, but need a confidential file on a server within your intranet. By obtaining Internet access in the form of a dial-in connection to an ISP such as IBM Global Services, you can communicate with the server in your intranet and access the required file.

One way to implement this scenario is to use an eNetwork VPN IPSec-enabled remote client and firewall, as shown in Figure 9 on page 22. The client accesses the Internet via dial-up to an ISP, and then establishes an authenticated and encrypted tunnel between itself and the firewall at the intranet boundary.

By applying IPSec authentication between the remote client and the firewall, you can protect your intranet from unwanted and possibly malicious IP packets. And by encrypting traffic that flows between the remote host and the firewall, you can prevent outsiders from eavesdropping on your information.

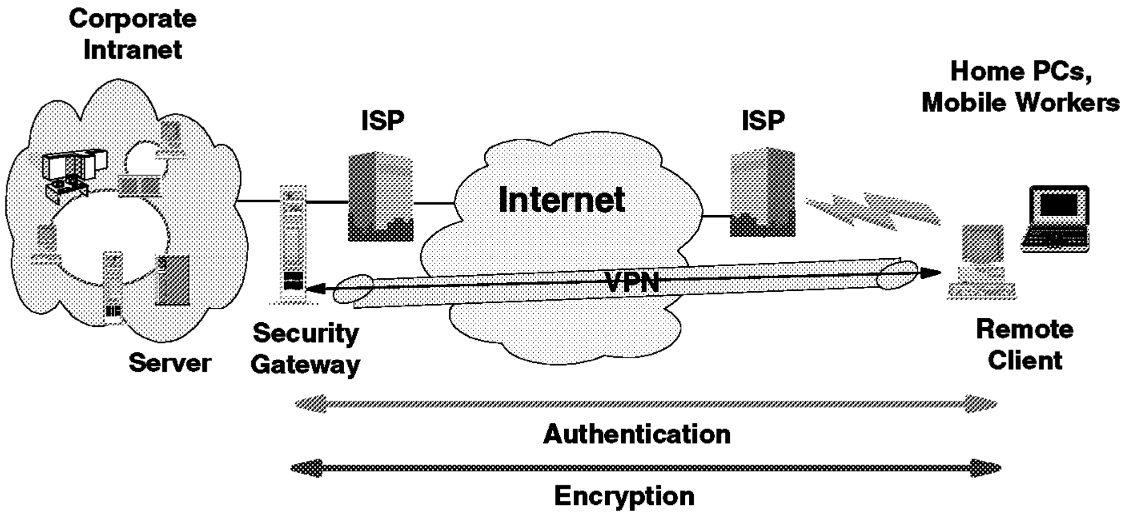


Figure 9. Remote Access Network

The three scenarios discussed in this section are the basis for the IPSec implementation and configuration examples described in this redbook. But before we come to the practical part, we would like to discuss the theory behind cryptography and IPSec in more detail.

Chapter 2. A Short Introduction to Cryptography

The purpose of this chapter is to introduce the terminology and give a brief overview of the major cryptographic concepts that relate to IPSec as a foundation of the virtual private networks. The pivot of any VPN technology is its cryptographic feature set. One should not plan and implement VPNs without knowing what level of security can be achieved with a given technology incorporated in a certain product. After all, you do not want to see your sensitive information at stake when crossing insecure channels.

The information presented here only scratches the surface. Some issues are left open or not mentioned at all. The more interested reader should consult the reference works listed in Appendix B, “Related Publications” on page 209.

2.1 Terminology

Let′s start with defining some very basic concepts.

Cryptography

Put simply, cryptography is the science of keeping your data and communications secure. To achieve this goal, techniques such as *encryption, decryption* and *authentication* are used. With the recent advances in this field, the frontiers of cryptography have become blurred. Every procedure consisting of transforming data based on methods that are difficult to reverse can be considered cryptography. The key factor to strong cryptography is the difficulty of reverse engineering. You would be amazed to know that breaking simple methods such as password-scrambled word processor documents or compressed archives is a matter of minutes for a hacker using an ordinary PC. *Strong* cryptography means that the computational effort needed to retrieve your cleartext messages without knowing the proper procedure makes the retrieval infeasible. In this context, infeasible means something like this: if all the computers in the world were assigned to the problem, they would have to work tens of thousands of years until the solution was found. The process of retrieval is called *cryptanalysis*. An attempted cryptanalysis is an *attack*.

Encryption and Decryption - Cryptographic Algorithms

Encryption is the transformation of a cleartext message into an unreadable form in order to hide its meaning. The opposite transformation, which retrieves the original cleartext, is the decryption. The mathematical function used for encryption and decryption is the *cryptographic algorithm* or *cipher*.

The security of a cipher might be based entirely on keeping how it works secret, in which case it is a *restricted* cipher. There are many drawbacks to restricted ciphers. It is very difficult to keep in secret an algorithm used by many people. If it is incorporated in a commercial product, then it is only a matter of time and money to get it reverse engineered. For these reasons, the currently used algorithms are *keyed*, that is, the encryption and decryption makes use of a parameter, the *key*. The key can be chosen from a set of possible values, called the *keyspace*. The keyspace usually is huge, the bigger the better. The security of these algorithms rely entirely on the key, not on their internal secrets. In fact the algorithms themselves are public and are extensively analyzed for possible weaknesses.

Note: As a general rule, be conservative. Do not trust brand new, unknown or unpublished algorithms. The principle of the keyed ciphers is shown in Figure 10 on page 24.

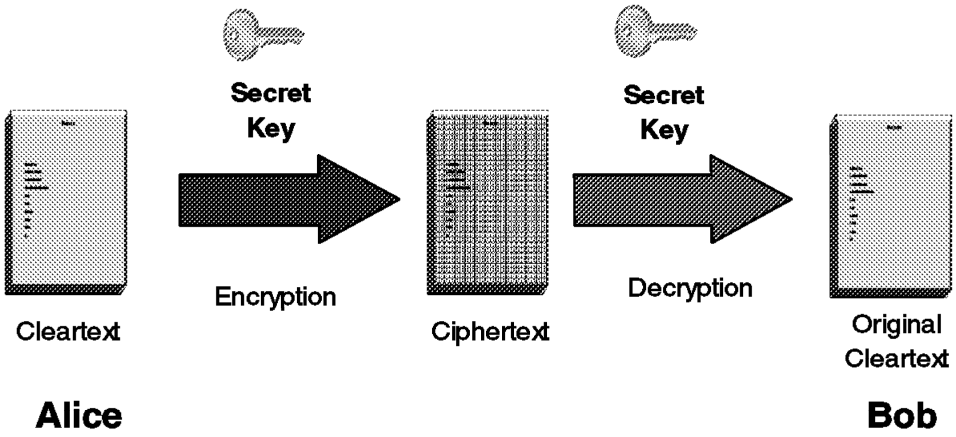


Figure 10. Keyed Encryption and Decryption

Note: It is common in the cryptographic literature to denote the first participant in a protocol as Alice and the second one as Bob. They are the ″crypto couple″.

Authentication, Integrity, and Non-repudiation

Encryption provides confidentiality to your messages. When communicating over an untrusted medium, such as the Internet, besides confidentiality, you need more:

- *Authentication* - A method for verifying that the sender of a message is really he or she claims to be. Any intruder masquerading as someone else is detected by authentication.
- *Integrity checking* - A method for verifying that a message has not been altered along the communication path. Any tampered message sent by an intruder is detected by integrity check. As a side effect, communication errors are also detected.
- *Non-repudiation* - The possibility to prove that the sender has really sent the message. When algorithms providing non-repudiation are used, the sender is not able to later deny the fact that he or she sent the message in question.

2.2 Symmetric or Secret-Key Algorithms

The symmetric algorithms are keyed algorithms where the decryption key is the same as the encryption key. These are the conventional cryptographic algorithms where the sender and the receiver must agree on the key *before* any secured communication can take place between them. Figure 10 illustrates a symmetric algorithm. There are two types of symmetric algorithms: *block algorithms*, which operate on the cleartext in blocks of bits, and *stream algorithms*, which operate on a single bit (or byte) of cleartext at a time.

Block ciphers are used in several *modes*. *Electronic Codebook Mode (ECB)* is the simplest; each block of cleartext is encrypted independently. Given a block length of 64 bits, there are 2^{64} possible input cleartext blocks, each of them corresponding to exactly one out of 2^{64} possible ciphertext blocks. An intruder might construct a codebook with known cleartext-ciphertext pairs and mount an attack. Because of this vulnerability, often the *Cipher Block Chaining (CBC)* mode is used, where the result of the encryption of the previous block is used in the encryption of the current block, thus each ciphertext block is dependent not just on the corresponding plaintext block, but on all previous plaintext blocks.

The algorithms often make use of *initialization vectors (IVs)*. These are variables independent of the keys and are good for setting up the initial state of the algorithms.

A well-known block algorithm is DES, a worldwide standard cipher developed by IBM. DES is an acronym for Data Encryption Standard. DES operates on 64-bit blocks and has a key length of 56 bits, often expressed as a 64-bit number, with every eighth bit serving as parity bit. From this key 16 subkeys are derived, which are used in the 16 rounds of the algorithm.

DES produces ciphertexts of the same length as the cleartext and the decryption algorithm is exactly the same as the encryption, the only difference being the subkey schedule. These properties makes it very suitable for hardware implementations.

Although DES is aging (its origins dates back to the early ′70s), after more then 20 years of analysis the algorithm itself is still considered secure. The most practical attack against it is *brute-force*: try the decryption with all possible keys and look for a meaningful result. The problem is the key length. Given enough money and time, a brute-force attack against the 56-bit key might be feasible; that′s why recently a new mode of DES, called triple-DES or 3DES has gained popularity. With triple-DES, the original DES algorithm is applied in three rounds, with two or three different keys. This encryption is thought to be unbreakable for a long time, even with the foreseeable technological advances taken into account.

An exportable version of DES is IBM′s Commercial Data Masking Facility or CDMF, which uses a 40-bit key.

Another, more recent block algorithm is the *International Data Encryption Algorithm (IDEA)*. This cipher uses 64-bit blocks and 128-bit keys. It was developed in the early ′90s and aimed to replace DES. It is cryptographically strong and faster than DES. Despite of this, there is no widespread commercial acceptance, mainly because it is relatively new and not fully analyzed. The most significant use of IDEA is in the freeware secure e-mail package Pretty Good Privacy (PGP).

An example of a stream algorithm is A5 which is used to encrypt digital cellular telephony traffic in the GSM standard, widely used in Europe.

The advantage of the symmetric algorithms is their efficiency. They can be easily implemented in hardware. A major disadvantage is the difficulty of key management. A secure way of exchanging the keys must exist, which is often very hard to implement.

2.2.1 Usage of Symmetric Keys with IPSec

These algorithms are used in the ESP protocol of the IPSec framework. Current specifications require only DES in CBC mode, but triple-DES (Internet Draft specification available) and CDMF (as a vendor-specific option with ISAKMP/Oakley) are also used. Specifications also exist for the usage of IDEA and some other encryption algorithms, however support for these is not widespread.

There are two variants of DES in CBC mode that are used with ESP:

- DES-CBC with a 64-bit initialization vector: The ESP protocol header carries the whole initialization vector. This variant is marked as DES_CBC_8 in the IBM products.
- DES-CBC with a 32-bit initialization vector: The ESP protocol header carries only a 32-bit value as IV, from which the full 64-bit IV is generated by concatenation of the original 32-bit IV and its complement. You can find this variant as DES_CBC_4 in the IBM products.

There is no significant difference in the security level of the variants. DES_CBC_8 has a slightly greater overhead. Most implementations, for example the IBM Firewall, use sequential IVs.

2.3 Asymmetric or Public-Key Algorithms

These algorithms address the major drawback of the symmetric ones, the requirement of the secure key-exchange channel. The idea is that two different keys should be used: a public key, which as the name implies, is known to everyone, and a private key, which is to be kept in tight security by the owner. The private key cannot be determined from the public key. A cleartext encrypted with the public key can only be decrypted with the corresponding private key, and vice versa. A cleartext encrypted with the private key can only be decrypted with the corresponding public key. Thus, if someone sends a message encrypted with the recipient′s public key, it can be read by the intended recipient only. The process is shown in Figure 11, where Alice sends an encrypted message to Bob.

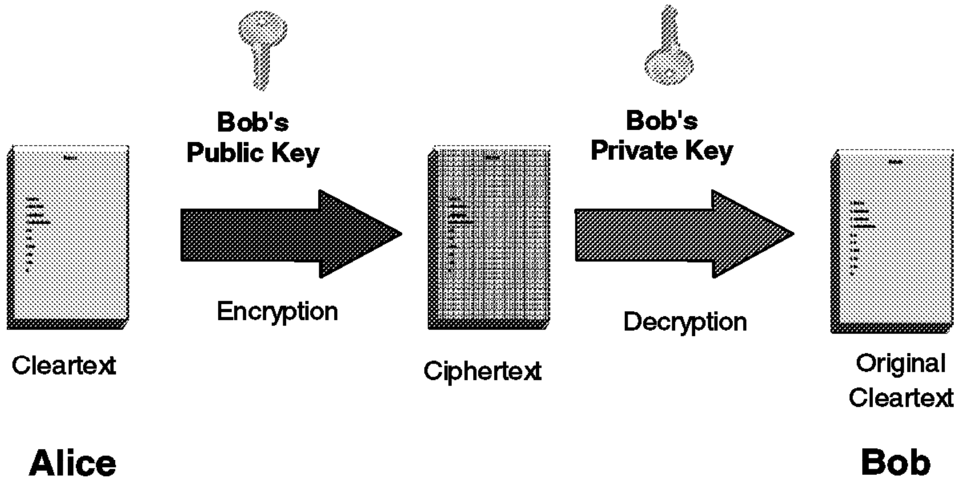


Figure 11. Encryption Using the Recipient′s Public Key

As the public key is available to anyone, privacy is assured without the need for a secure key-exchange channel. Parties who wish to communicate retrieve each other′s public key.

2.3.1 Authentication and Non-Repudiation

An interesting property of the public-key algorithms is that they can provide authentication. Use the private key for encryption. Since anyone has access to the corresponding public key and can decrypt the message, this provides no privacy. However, it authenticates the message. If one can successfully decrypt it with the claimed sender′s public key, then the message has been encrypted with the corresponding private key, which is known by the real sender only. Thus, the sender′s identity is verified. The encryption with the private key is used in *digital signatures*. In Figure 12 the principle is shown. Alice encrypts her message with her private key (″signs″ it), in order to enable Bob to verify the authenticity of the message.

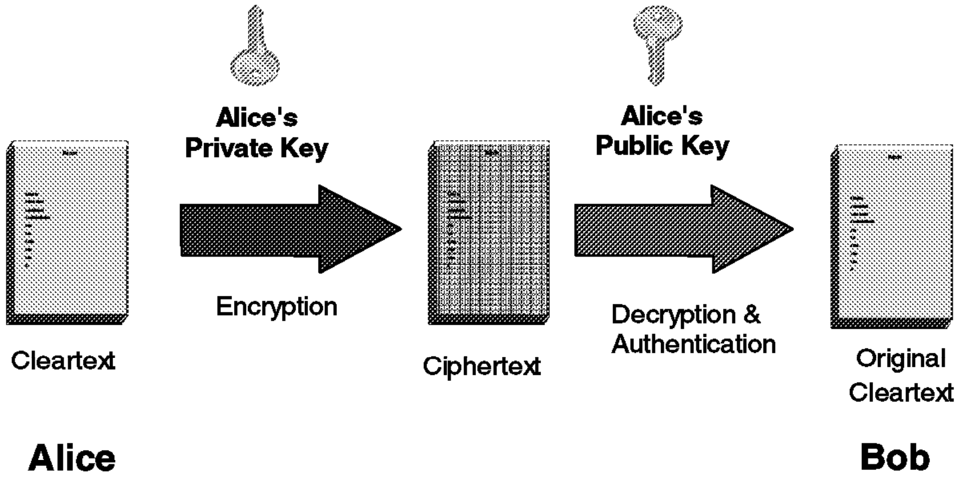


Figure 12. Authentication by Encrypting with a Private Key

Going a step forward, encrypting with the private key gives non-repudiation too. The mere existence of such an encrypted message testifies that the originator has really sent it, because only he or she could have used the private key to generate the message. Additionally, if a timestamp is included, then the exact date and time can also be proven. There are protocols involving trusted third parties that prevent the sender from using phony timestamps.

Note: Inspired by the ″stamping″ idea, the IPSec architecture makes use of sequence numbers (instead of timestamps), to achieve replay protection.

2.3.2 Examples of Public-Key Algorithms

Algorithms based on public keys can be used for a variety of purposes. Two common applications are:

1. Encryption (see 2.3.3.1, “RSA Public Key Algorithm” on page 28)
2. Generation of shared keys for use with symmetric key algorithms (see 2.3.3.2, “Diffie-Hellman Key Exchange” on page 29)

The most popular public-key algorithm is the de-facto standard *RSA*, named after the three inventors: Ron Rivest, Adi Shamir and Leonard Adleman. The security of RSA relies on the difficult problem of factoring large numbers. The public and private keys are functions of two very large (200 digits or even more) prime numbers. Given the public key and the ciphertext, an attack would be successful if it could factor the product of the two primes. RSA has resisted many years of extensive attacks. As computing power grows, keeping RSA secure is a matter of increasing the key length. (As opposed to DES, where the key length is fixed.)

Another public-key algorithm, actually the very first ever invented, is *Diffie-Hellman*. This is a key-exchange algorithm, that is, it is used for securely establishing a shared secret over an insecure channel. The communicating parties exchange public information from which they derive a key. An eavesdropper cannot reconstruct the key from the information that went through the insecure channel. (More precisely, the reconstruction is computationally infeasible.) The security of Diffie-Hellman relies on the difficulty of calculating discrete logarithms in finite fields. After the shared secret has been established, it can then be used to derive keys for use with symmetric key algorithms such as DES.

Diffie-Hellman makes possible the secure derivation of a shared secret key, but it does not authenticate the parties. For authentication another public-key algorithm must be used, such as RSA.

Unfortunately, public-key algorithms while providing for easier key management, privacy, authentication and non-repudiation also have some disadvantages. The most important one is that they are slow and difficult to implement in hardware. For example, RSA is 100 to 10000 times slower than DES, depending on implementation. Because of this, public-key algorithms generally are not used for bulk encryption. Their most important use is key exchange and authentication. Another notable disadvantage is that they are susceptible to certain cryptanalytic attacks to which symmetric algorithms are resistant.

Therefore, a good cryptographic system (*cryptosystem*) makes use of both worlds. It uses public-key algorithms in the session establishment phase for authentication and key exchange, then a symmetric one for encrypting the consequent messages.

2.3.3 Usage of Asymmetric Keys with IPSec

IPSec uses asymmetric algorithms for secure key generation and authentication. These operations are typical in the ISAKMP/Oakley framework.

For the interested reader, below we give more detailed information of the two most important asymmetric algorithms. Both of them involve modular arithmetics. An arithmetic operation modulo m means that the result of that operation is divided by m and the remainder is taken. For example: 3 * 6 mod 4 = 2, since 3 * 6 = 18 and dividing 18 by 4 gives us 2 as the remainder.

2.3.3.1 RSA Public Key Algorithm

RSA is used in the ISAKMP/Oakley framework as one of the possible authentication methods. The principle of the RSA algorithm is as follows:

1. Take two large primes, p and q.
2. Find their product n = pq; n is called the modulus.
3. Choose a number, e, less than n and relatively prime to (p-1)(q-1) which means that e and (p-1)(q-1) have no common factor other than 1.
4. Find its inverse, d mod (p-1)(q-1) which means that ed = 1 mod (p-1)(q-1).

e and d are called the public and private exponents, respectively. The public key is the pair (n,e); the private key is d. The factors p and q must be kept secret or destroyed.

A simplified example of RSA encryption would be the following:

1. Suppose Alice wants to send a private message, m, to Bob. Alice creates the ciphertext c by exponentiating:

   $c = m^e \bmod n$

   where e and n are Bob′s public key.

2. Alice sends c to Bob.

3. To decrypt, Bob exponentiates:

   $m = c^d \bmod n$

   and recovers the original message; the relationship between e and d ensures that Bob correctly recovers m. Since only Bob knows d, only Bob can decrypt the ciphertext.

A simplified example of RSA authentication would be the following:

1. Suppose Alice wants to send a signed message, m, to Bob. Alice creates a digital signature s by exponentiating:

   $s = m^d \bmod n$

   where d and n belong to Alice′s private key.

2. She sends s and m to Bob.

3. To verify the signature, Bob exponentiates and checks if the result, compares to m:

   $m = s^e \bmod n$

   where e and n belong to Alice′s public key.

2.3.3.2 Diffie-Hellman Key Exchange

The Diffie-Hellman key exchange is a crucial component of the ISAKMP/Oakley framework. In the earliest phase of a key negotiation session there is no secure channel in place. The parties derive shared secret keys using the Diffie-Hellman algorithm. These keys will be used in the next steps of the key negotiation protocol.

The outline of the algorithm is the following:

1. The parties (Alice and Bob) share two public values, a modulus m and an integer g; m should be a large prime number.

2. Alice generates a large random number a and computes:

   $X = g^a \bmod m$

3. Bob generates a large random number b and computes:

   $Y = g^b\bmod m$

4. Alice sends X to Bob.

5. Bob computes:

   $K1 = X^b \bmod m$

6. Bob sends Y to Alice.

7. Alice computes:

   $K2 = Y^a \bmod m$

Both K1 and K2 are equal to g^{ab} mod m. This is the shared secret key. Noone is able to generate this value without knowing a or b. The security of the exchange is based on the fact that is extremely difficult to inverse the exponentiation performed by the parties. (In other words, to find out discrete logarithms in finite fields of size m.) Similar to RSA, advances in adversary computing power can be countered by choosing larger initial values, in this case a larger modulus m.

Please see Chapter 10, “The Internet Key Exchange (IKE) Protocol” on page 193 for more details on how ISAKMP/Oakley uses Diffie-Hellman exchanges.

2.4 Hash Functions

Hash functions (also called message digests) are fundamental to cryptography. A hash function is a function that takes variable-length input data and produces fixed length output data (the hash value), which can be regarded as the ″fingerprint″ of the input. That is, if the hashes of two messages match, than we get a high assurance that the messages are the same.

Cryptographically useful hash functions must be *one-way*, which means that they should be easy to compute, but infeasible to reverse. An everyday example of a one-way function is mashing a potato; it its easy to do, but once mashed, reconstructing the original potato is rather difficult. A good hash function should be *collision-resistant*. It should be hard to find two different inputs that hash to the same value. As any hash function maps an input set to a smaller output set, theoretically it is possible to find collisions. The point is to provide a unique digital ″fingerprint″ of the message, that identifies it with high confidence, much like a real fingerprint identifying a person.

A hash function that takes a key as a second input parameter and its output depends on both the message and the key is called a *Message Authentication Code (MAC)*, as shown in Figure 13.

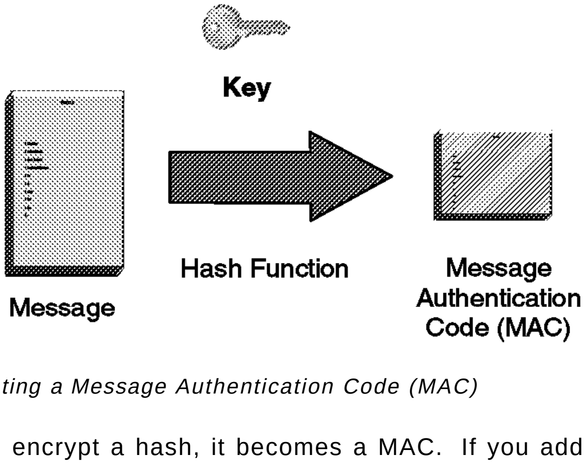


Figure 13. Generating a Message Authentication Code (MAC)

Put simply, if you encrypt a hash, it becomes a MAC. If you add a secret key to a message, then hash the concatenation, the result is a MAC. Both symmetric an asymmetric algorithms can be used to generate MACs.

Hash functions are primarily used in integrity check and authentication techniques. Let′s see how integrity and authentication is assured with hash functions:

- The sender calculates the hash of the message and appends it to the message.
- The recipient calculates the hash of the received message and then compares the result with the transmitted hash.
- If the hashes match, the message was not tampered with.
- In case of MACs where the encryption key (symmetric or asymmetric) should have been used by a trusted sender only, a successful MAC decryption indicates that the claimed and actual senders are identical. (Unless, of course, your keys have been compromised.)

See Figure 14 for an illustration of the procedure. The Message* and MAC* notations reflect the fact that the message might have been altered while crossing the untrusted channel.

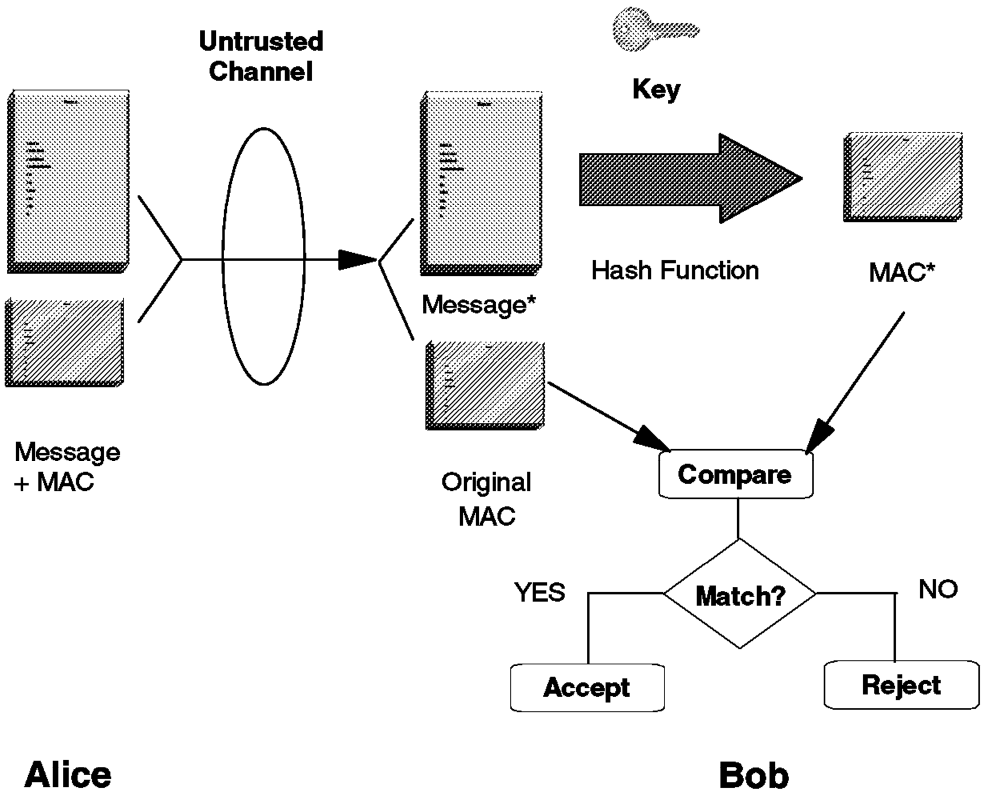


Figure 14. Checking Integrity and Authenticity with MAC

One could argue that the same result can be obtained with any kind of encryption, because if an intruder modifies an encrypted message, the decryption will result in nonsense, thus tampering can be detected. The answer is that many times only integrity and/or authentication is needed, maybe with encryption on some of the fields of the message. And encryption is very processor-intensive. (Examples are the personal banking machine networks, where only the PINs are encrypted, however MACs are widely used. Encrypting all the messages in their entirety would not yield noticeable benefits and performance would dramatically decrease.)

The encryption of a hash with the private key is called a *digital signature*. It can be thought of as a special MAC. Using digital signatures instead of encrypting the whole message with the private key leads to considerable performance gains and a remarkable new property. The authentication part can be decoupled from the document itself. This property is used for example in the Secure Electronic Transaction (SET) protocol.

The encryption of a secret key with a public key is called a *digital envelope*. This is a common technique used to distribute secret keys for symmetric algorithms.

Note: In the IPSec vocabulary, the distinct functions of authentication, integrity and replay protection are commonly referred to as authentication. Generally MAC is referred to as authentication data or integrity check value (ICV).

2.4.1 Examples of Hash Functions

The most widely used hash functions are MD5 and Secure Hash Algorithm 1 (SHA-1). MD5 was designed by Ron Rivest (co-inventor of RSA). SHA-1 is largely inspired from MD5 and was designed by the National Institute of Standards and Technology (NIST) and the National Security Agency (NSA) for use with the Digital Signature Standard (DSS). MD5 produces a 128-bit hash, while SHA-1 produces a 160-bit hash. Both functions encode the message length in their output. SHA-1 is regarded as more secure, because of the larger hashes it produces.

Note: Neither MD5 nor SHA-1 takes a key as input parameter, hence in their original implementation they cannot be used for MAC calculation. However, for this purpose it is easy to concatenate a key with the input data and apply the function to the result. In practice, for example in IPSec, often more sophisticated schemes are used.

2.4.2 Usage of Hash Functions with IPSec

The IPSec framework can use both MD5 and SHA-1 for MAC calculation to provide authentication and integrity. There are two ways of using each of the functions, which results in four different possibilities: Keyed MD5, Keyed SHA-1, HMAC-MD5-96 and HMAC-SHA-1-96. Other hash functions can also be accommodated.

2.4.2.1 Keyed MD5 and Keyed SHA-1

Using MD5 and SHA-1 in keyed mode is simple. The shared secret key and the datagram to be protected are both input to the hash algorithm and the output (the hash value) is placed in the Authentication Data field of the AH Header, as it is shown in Figure 15 on page 33.

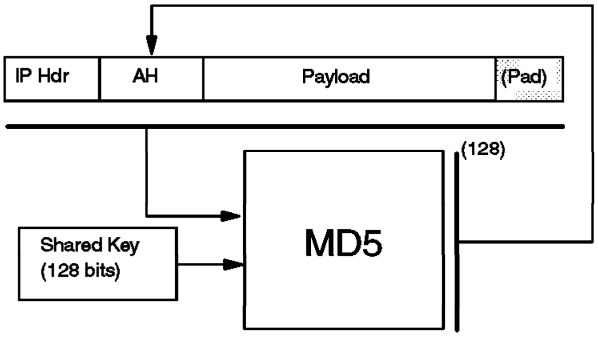


Figure 15. Keyed MD5 Processing

Keyed SHA-1 operates in exactly the same way, the only difference being the larger 160-bit hash value.

2.4.2.2 HMAC-MD5-96 and HMAC-SHA-1-96

A stronger method is the Hashed Message Authentication Code (HMAC), proposed by IBM. HMAC itself is not a hash function, rather a cryptographically strong way to use a specific hash function for MAC calculation.

Here is how HMAC works, considering MD5 as an example. The base function is applied twice in succession. In the first round the input to MD5 is the shared secret key and the datagram. The 128-bit output hash value and the key is input again to the hash function in the second round. The leftmost 96 bits of the resulting hash value is used as the MAC for the datagram. See Figure 16 for an illustration.

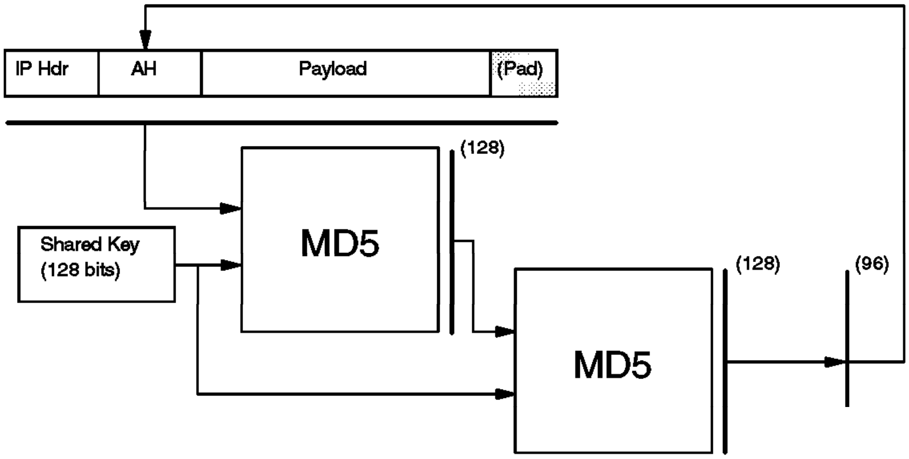


Figure 16. HMAC-MD5-96 Processing

HMAC-SHA-1-96 operates in the same way, except that the intermediary results are 160 bits long.

2.4.2.3 Digital Signature Standard (DSS)

As mentioned previously, a hash value encrypted with the private key is called a *digital signature* and is illustrated in Figure 17 on page 34.

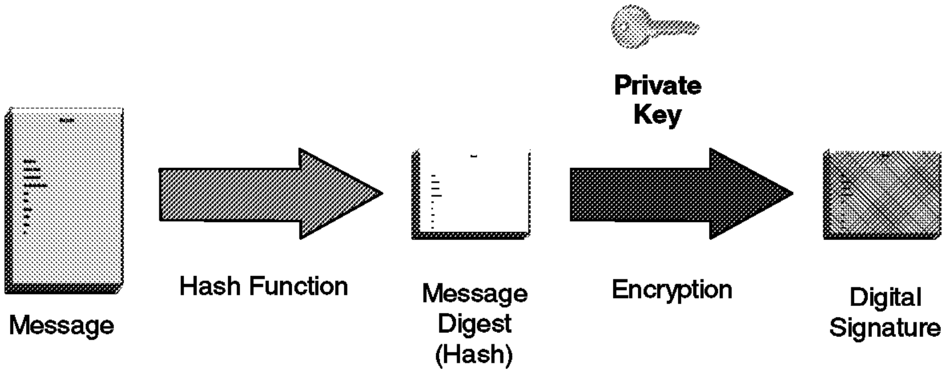


Figure 17. Generating a Digital Signature

One authentication method that can be used with ISAKMP/Oakley is DSS which was selected by NIST and NSA to be the digital authentication standard of the U.S. government. The standard describes the Digital Signature Algorithm (DSA) used to sign and verify signatures of message digests produced with SHA-1.

A brief description of DSA is given below:

1. Choose a large prime number, p, usually between 512 and 1024 bits long.
2. Find a prime factor q of (p-1), 160 bits long.
3. Compute:

   $g=h^{(p-1)/q} \bmod p$

   where h is a number less than (p-1) and the following is true:

   $h^{(p-1)/q}>1$
4. Choose another number x, less than q, as the sender′s private key.
5. Compute:

   $y=g^{x} \bmod p$

   and use that as the sender′s public key. The pair (x,y) is sometimes referred to as the long-term key pair.
6. The sender signs the message as follows:
   a. Generate a random number, k, less than q.
   b. Compute:

      $r=(g^{k} \bmod p) \bmod q$

      $s=(k^{-1}(SHA1(m)+xr)) \bmod q$

      The pair (k,r) is sometimes referred to as the per-session key pair, and the signature is represented by the pair (r,s).
7. The sender sends (m,r,s).
8. The receiver verifies the signature as follows:
   a. Compute:

      $w=s^{-1} \bmod q$

      $u1=(SHA1(m)*w) \bmod q$

      $u2=(rw) \bmod q$

$$v=((g^{u1}y^{u2}) \bmod p) \bmod q$$

9. If $v=r$, then the signature is verified.

The above description shows the principles of using the hash functions in IPSec. Details such as what fields to include in the calculations and how are omitted. See Chapter 3, “Description of IPSec” on page 39 for a thorough presentation of the inner workings of IPSec.

2.5 Digital Certificates and Certification Authorities

As we said before in 2.3.1, “Authentication and Non-Repudiation” on page 27, with public-key cryptography, the parties retrieve each other′s public key. There are security exposures here. An intruder could change some real public keys with his or her own public key, and then mount a so-called *man-in-the-middle attack*. It works like this. The intruder places himself between Alice and Bob. He can trick Bob by sending him one of his own public keys as if it were Alice′s. The same applies to Alice. She thinks she uses Bob′s public key, but the sour reality is that she actually uses the intruder′s. So the clever intruder can decrypt the confidential traffic between the two and remain undetected. For example, a message sent by Alice and encrypted with ″Bob′s″ public key lands at the intruder, who decrypts it, learns its content, then re-encrypts it with Bob′s real public key. Bob has no way to realize that Alice is using a phony public key.

An intruder could also use impersonation, claiming to be somebody else, for example an online shopping mall, fouling innocent shoppers.

The solution to these serious threats is the *digital certificate*. A digital certificate is a file that binds an identity to the associated public key. This binding is validated by a trusted third party, the *certification authority (CA)*. A digital certificate is signed with the private key of the certification authority, so it can be authenticated. It is only issued after a verification of the applicant. Apart the public key and identification, a digital certificate usually contains other information too, such as:

- Date of issue
- Expiry date
- Miscellaneous information from issuing CA (for example, serial number)

Note: There is an international standard in place for digital certificates: the ISO X.509 protocols.

Now the picture looks different. The parties retrieve each other′s digital certificate and authenticate it using the public key of the issuing certification authority. They have confidence that the public keys are real, because a trusted third party vouches for them. The malicious online shopping mall is put out of business.

It easy to imagine that one CA can not cover all needs. What happens when Bob′s certificate is issued by a CA unknown to Alice? Can she trust that unknown authority? Well, this is entirely her decision, but to make life easier, CAs can form a hierarchy, often referred to as the *trust chain*. Each member in the chain has a certificate signed by it superior authority. The higher the CA is in the chain, the tighter security procedures are in place. The root CA is trusted by everyone and its private key is real top secret.

Alice can traverse the chain upwards until she finds a CA that she trusts. The traversal consists of verifying the subordinate CA′s public key and identity using the certificate issued to it by the superior CA.

When a trusted CA is found up in the chain, Alice is assured that Bob′s issuing CA is trustworthy. In fact this is all about delegation of trust. We trust your identity card if somebody who we trust signs it. And if the signer is unknown to us, we can go upward and see who signs for the signer, etc.

An implementation of this concept can be found in the SET protocol, where the major credit card brands operate their own CA hierarchies that converge to a common root. Lotus Notes authentication, as another example, is also based on certificates, and it can be implemented using hierarchical trust chains. PGP also uses a similar approach, but its trust chain is based on persons and it is rather a distributed Web than a strict hierarchical tree.

2.5.1 Usage of Digital Certificates with IPSec

IPSec uses digital certificates in the ISAKMP negotiations, for the following authentication modes:

- Digital signature (DSS)
- RSA encryption
- RSA signature

An IPSec certificate has a named subject (the identity), which could be any of the following:

- IP address
- IP address range
- Subnet address
- Domain name
- Fully qualified domain name
- Distinguished name
- Text string

Please refer to Chapter 10, “The Internet Key Exchange (IKE) Protocol” on page 193 for a more detailed description of ISAKMP/Oakley.

2.6 Random-Number Generators

An important component of a cryptosystem is the random-number generator. Many times random session keys and random initialization variables (often referred to as initialization vectors) are generated. For example, DES requires an explicit initialization vector and Diffie-Hellman relies on picking random numbers which serve as input for the key derivation.

The quality, that is the randomness of these generators is more important than you would think. The ordinary random function provided with most programming language libraries is good enough for games, but not for cryptography. Those random-number generators are rather predictable; if you rely on them, be prepared for happy cryptanalysts finding interesting correlations in your encrypted output.

The fundamental problem faced by the random-number generators is that the computers are ultimately deterministic machines, so real random sequences cannot be produced. As John von Neumann ironically said: ″Anyone who considers arithmetical methods of producing random digits is, of course, in a state of sin″. (Quoted by Donald Knuth.) That′s why the term *pseudorandom generator* is more appropriate.

Cryptographically strong pseudorandom generators must be unpredictable. It must be computationally infeasible to determine the next random bit, even with total knowledge of the generator.

A common practical solution for pseudorandom generators is to use hash functions. This approach provides sufficient randomness and it can be efficiently implemented. Military-grade generators use specialized devices that exploit the inherent randomness in physical phenomena. An interesting solution can be found in the PGP software. The initial seed of the pseudorandom generator is derived from measuring the time elapsed between the keystrokes of the user.

Note: The IPSec specifications state that a strong random-number generator must be used for initialization vector and key generation.

2.7 Export/Import Restrictions on Cryptography

U.S. export regulations changed in 1996 that put cryptography under the control of the Commerce Department. It had formerly been treated as a munition. This is a significant step in liberalizing the export of cryptographic products.

According to the new export regulations a license may be granted to export a 56-bit key encryption algorithm if a company has an approved key recovery plan. The key recovery plan must be implemented in 2 years and the license is granted on a 6 month basis.

In 1997 IBM has been granted the license to export DES as long as its use is similar to other products that have been approved. Recently, the export of triple-DES has been allowed for banking applications.

In France, according to the law, any product capable of enciphering/deciphering user data should be granted a license from the French government before being marketed. Then customers need to be authorized to use them on a case-by-case basis. In reality, two major and useful exceptions exist:

1. Routinely, licenses are granted that allow banks to use DES products on a global basis (no case-by-case authorization required).
2. Routinely, global licenses are granted that allow anybody to use weak encryption (RC2/RC4 with 40-bit keys).

Chapter 3. Description of IPSec

In this chapter we examine in detail the IPSec framework, its two main protocols, Authentication Header (AH) and Encapsulating Security Payload (ESP). The header formats, the specific cryptographic features and the different modes of application of AH and ESP are discussed.

Note: The third IPSec component, the Internet Key Exchange (IKE), formerly referred to as ISAKMP/Oakley, is mentioned briefly in Chapter 10, "The Internet Key Exchange (IKE) Protocol" on page 193. This decision is based upon the currently available IBM IPSec implementations that do no yet support IKE. IKE and its support within IBM products are discussed in detail in a separate redbook to be published early next year.

IPSec was designed for interoperability. When correctly implemented, it does not affect networks and hosts that do not support it. IPSec is independent of the current cryptographic algorithms, it can accommodate new ones as they become available. It works both with IPv4 and IPv6. Actually IPSec is a mandatory component of IPv6.

IPSec uses state-of-the-art cryptographic algorithms. The specific implementation of an algorithm for use by an IPSec protocol is often called a *transform*. For example, the DES algorithm used in ESP is called the ESP DES-CBC transform. The transforms, as the protocols, are published in RFCs and in Internet Drafts.

Note: Internet Drafts are working documents and are valid for a maximum of 6 months. You should check the Internet Engineering Task Force document repository (http://www.ietf.org/home.html) or another up-to-date repository for the latest drafts.

3.1 Concepts

Two major IPSec concepts should be clarified before entering the details: the Security Associations and the tunneling. In fact they are not new, IPSec just makes use of them. These concepts are described in the following sections.

3.1.1 Security Associations

The concept of a Security Association (SA) is fundamental to IPSec. An SA is a unidirectional (simplex) logical connection between two IPSec systems, uniquely identified by the following triple:

```
<Security Parameter Index, IP Destination Address, Security Protocol>
```

The definition of the members is as follows:

Security Parameter Index (SPI)

This is a 32-bit value used to identify different SAs with the same destination address and security protocol. The SPI is carried in the header of the security protocol (AH or ESP). The SPI has only local significance, as defined by the creator of the SA. The SPI values in the range 1 to 255 are reserved by the Internet Assigned Numbers Authority (IANA). The SPI value of 0 must be used for local implementation-specific purposes only. Generally the SPI is selected by the destination system during the SA establishment.

IP Destination Address

This address may be a unicast, broadcast or multicast address. However, currently SA management mechanisms are defined only for unicast addresses.

Security Protocol

This can be either AH or ESP.

An SA can be in either of two modes: transport or tunnel, depending on the mode of the protocol in that SA. You can find the explanation of these protocol modes later in this chapter.

Because SAs are simplex, for bidirectional communication between two IPSec systems, there must be two SAs defined, one in each direction.

An SA gives security services to the traffic carried by it either by using AH or ESP, but not both. In other words, for a connection that should be protected by both AH and ESP, two SAs must be defined for each direction. In this case, the set of SAs that define the connection is referred to as an *SA bundle*. The SAs in the bundle do not have to terminate at the same endpoint. For example, a mobile host could use an AH SA between itself and a firewall and a nested ESP SA that extends to a host behind the firewall.

An IPSec implementation maintains two databases related to SAs:

Security Policy Database (SPD)

The Security Policy Database specifies what security services are to be offered to the IP traffic, depending on factors such as source, destination, whether it is inbound, outbound, etc. It contains an ordered list of policy entries, separate for inbound and or outbound traffic. These entries might specify that some traffic must not go through IPSec processing, some must be discarded and the rest must be processed by the IPSec module. Entries in this database are similar to the firewall rules or packet filters.

Security Association Database (SAD)

The Security Association Database contains parameter information about each SA, such as AH or ESP algorithms and keys, sequence numbers, protocol mode and SA lifetime. For outbound processing, an SPD entry points to an entry in the SAD. That is, the SPD determines which SA is to be used for a given packet. For inbound processing, the SAD is consulted to determine how the packet must be processed.

Notes:

1. The user interface of an IPSec implementation usually hides or presents in a more friendly way these databases and makes the life of the administrator easier.
2. IPSec policies will be discussed in more detail in the previously mentioned redbook about IKE, to be published at the end of this year.

3.1.2 Tunneling

Tunneling or encapsulation is a common technique in packet-switched networks. It consists of wrapping a packet in a new one. That is, a new header is attached to the original packet. The entire original packet becomes the payload of the new one, as it is shown in Figure 18 on page 41.

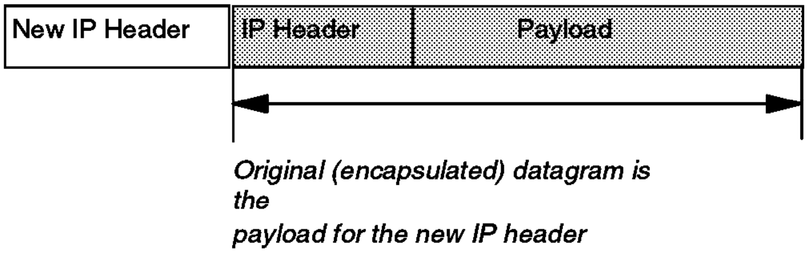

New IP Header
IP Header
Payload
Original (encapsulated) datagram is the
payload for the new IP header

Figure 18. IP Tunneling

In general tunneling is used to carry traffic of one protocol over a network that does not support that protocol directly. For example, NetBIOS or IPX can be encapsulated in IP to carry it over a TCP/IP WAN link. In the case of IPSec, IP is tunneled through IP for a slightly different purpose: to provide total protection, including the header of the encapsulated packet. If the encapsulated packet is encrypted, an intruder cannot figure out for example the destination address of that packet. (Without tunneling he or she could.) The internal structure of a private network can be concealed in this way.

Tunneling requires intermediate processing of the original packet on its route. The destination specified in the outer header, usually an IPSec firewall or router, retrieves the original packet and sends it to the ultimate destination. The processing overhead is compensated by the extra security.

A notable advantage of IP tunneling is the possibility to exchange packets with private IP addresses between two intranets over the public Internet, which requires globally unique addresses. Since the encapsulated header is not processed by the Internet routers, only the endpoints of the tunnel (the gateways) have to have globally assigned addresses; the hosts in the intranets behind them can be assigned private addresses, for example 10.x.x.x. As globally unique IP addresses are becoming a scarce resource, this interconnection method gains importance.

Note: IPSec tunneling is modeled after RFC 2003 ″IP Encapsulation within IP″. It has originally been designed for Mobile IP, an architecture that allows a mobile host to keep its home IP address even if attached to remote or foreign subnets.

3.1.3 Terminology Used throughout IPSec Redbooks

IPSec is a relatively new technology and it has a less coherent terminology than IP in general. In this section we summarize how the IPSec terms are used by us.

Gateway, Router and Firewall

Although these are separate entities, often they can be used interchangeably when the IPSec functionality is in focus. Usually we use the term gateway to denote a machine which routes IP traffic, as opposed to a host, which generates or consumes that traffic. The term *security gateway* is analogous. It is more precise since the name implies that the box is IPSec-capable.

IPSec Tunnel

This term is used to denote a pair of SAs that realize a bidirectional connection between two IPSec systems. It does not imply either transport or tunnel mode. Sometimes it is called simply a *tunnel*.

Selectors
Selectors define the IPSec processing of the outbound packets. The SPD entries consist of one or more selectors.

Packet Filters
These are rules that steer traffic into or out of the tunnel. The traffic might be either inbound or outbound.

3.2 Authentication Header (AH)

AH is used to provide integrity and authentication to IP datagrams. Optional replay protection is also possible. Although its usage is optional, the replay protection service must be implemented by any IPSec-compliant system. The mentioned services are connectionless, that is they work on a per-packet basis.

AH authenticates as much of the IP datagram as possible. Some fields in the IP header change en-route and their value cannot be predicted by the receiver. These fields are called *mutable* and are not protected by AH. The mutable IPv4 fields are:

- Type of Service (TOS)
- Flags
- Fragment Offset
- Time to Live (TTL)
- Header Checksum

When protection of these fields is required, tunneling should be used. The payload of the IP packet is considered immutable and is always protected by AH.

AH is identified by protocol number 51, assigned by the IANA. The protocol header (IPv4, IPv6, or Extension) immediately preceding the AH header contains this value in its Protocol (IPv4) or Next Header (IPv6, Extension) field.

AH processing is applied only to non-fragmented IP packets. However an IP packet with AH applied can be fragmented by intermediate routers. In this case the destination first reassembles the packet and then applies AH processing to it. If an IP packet that appears to be a fragment (offset field is non-zero, or the More Fragments bit is set) is input to AH processing, it is discarded. This prevents the so-called *overlapping fragment attack*, which misuses the fragment reassembly algorithm in order to create forged packets and force them through a firewall.

Packets that failed authentication are discarded and never delivered to upper layers. This mode of operation greatly reduces the chances of successful *denial of service* attacks, which aim to block the communication of a host or gateway by flooding it with bogus packets.

3.2.1 AH Header Format

The current AH header format is described in the Internet Draft *draft-ietf-ipsec-auth-header-06.txt*, which contains important modifications compared to the previous AH specification, RFC 1826. The information in this section is based on the respective Internet Draft.

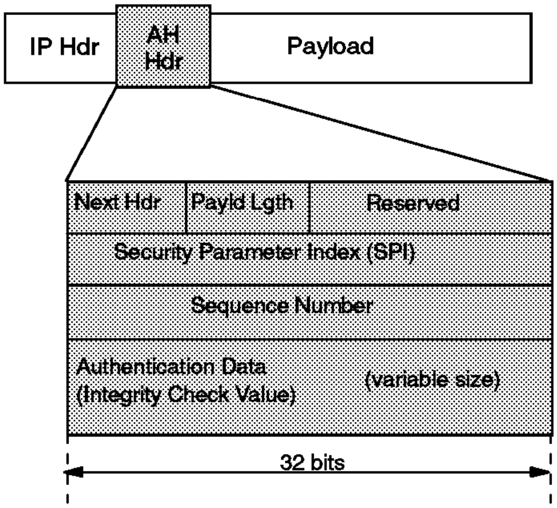


Figure 19. AH Header Format

In Figure 19 the position of the AH header in the IP packet and the header fields are shown. The explanation of the fields are as follows:

Next Header
The Next Header is an 8-bit field that identifies the type of the next payload after the Authentication Header. The value of this field is chosen from the set of IP protocol numbers defined in the most recent ″Assigned Numbers″ RFC from the Internet Assigned Numbers Authority (IANA).

Payload Length
This field is 8 bits long and contains the length of the AH header expressed in 32-bit words, minus 2. It does not relate to the actual payload length of the IP packet as a whole. If default options are used, the value is 4. (Three 32-bit fixed words plus three 32-bit words of authentication data minus two.)

Reserved
This field is reserved for future use. Its length is 16 bits and it is set to zero.

Security Parameter Index (SPI)
This field is 32 bits in length. See Security Parameter Index (SPI) on page 39 for a definition.

Sequence Number
This 32-bit field is a monotonically increasing counter which is used for replay protection. Replay protection is optional; however, this field is mandatory. The sender always includes this field and it is at the discretion of the receiver to process it or not. At the establishment of an SA the sequence number is initialized to zero. The first packet transmitted using the SA has a sequence number of 1. Sequence numbers are not allowed to repeat. Thus the maximum number of IP packets that can be transmitted on any given SA is 2^{32}-1. After the highest sequence number is used, a new SA and consequently a new key is established. Anti-replay is enabled at the sender by default. If upon SA establishment the receiver chooses not to use it, the sender does not concern with the value in this field anymore.

Notes:

1. Typically the anti-replay mechanism is not used with manual key management.
2. The original AH specification in RFC 1826 did not discuss the concept of sequence numbers. Older IPSec implementations that are based on that RFC can therefore not provide replay protection.

Authentication Data

This is a variable-length field, also called Integrity Check Value (ICV). The ICV for the packet is calculated with the algorithm selected at the SA initialization. The authentication data length is an integral multiple of 32 bits. As its name tells, it is used by the receiver to verify the integrity of the incoming packet.

In theory any MAC algorithm can be used to calculate the ICV. The specification requires that HMAC-MD5-96 and HMAC-SHA-1-96 must be supported. The old RFC 1826 requires Keyed MD5. In practice Keyed SHA-1 is also used. Implementations usually support two to four algorithms.

When doing the ICV calculation, the mutable fields are considered to be filled with zero.

3.2.2 Ways of Using AH

AH can be used in two ways: transport mode and tunnel mode.

3.2.2.1 AH in Transport Mode

In this mode the original IP datagram is taken and the AH header is inserted right after the IP header, as it is shown in Figure 20. If the datagram already has IPSec header(s), then the AH header is inserted before any of those.

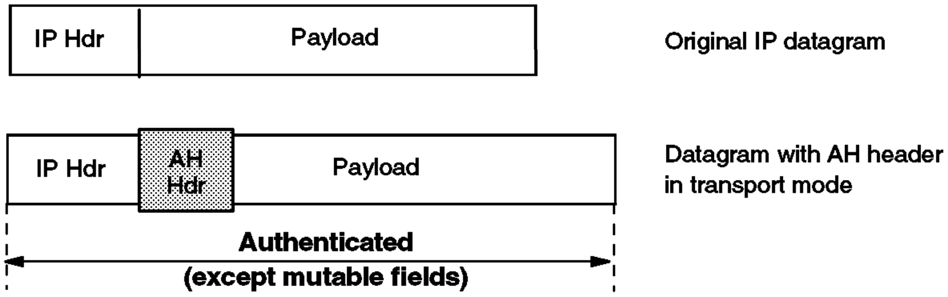

IP Hdr
Payload
Original IP datagram
IP Hdr
AH Hdr
Payload
Datagram with AH header in transport mode
Authenticated (except mutable fields)

Figure 20. Authentication Header in Transport Mode

The transport mode is used by hosts, not by gateways. Gateways are not even required to support transport mode.

The advantage of the transport mode is less processing overhead. The disadvantage is that the mutable fields are not authenticated.

3.2.2.2 AH in Tunnel Mode

With this mode the tunneling concept is applied a new IP datagram is constructed and the original IP datagram is made the payload of it. Then AH in transport mode is applied to the resulting datagram. See Figure 21 on page 45 for an illustration.

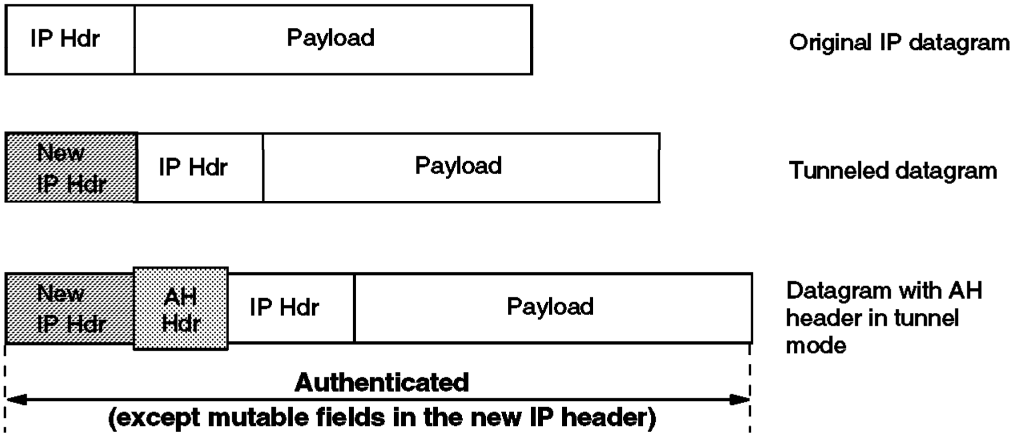


Figure 21. Authentication Header in Tunnel Mode

The tunnel mode is used whenever either end of a security association is a gateway. Thus, between two firewalls the tunnel mode is always used.

Although gateways are supposed to support tunnel mode only, often they can also work in transport mode. This mode is allowed when the gateway acts as a host, that is in cases when traffic is destined to itself. Examples are SNMP commands or ICMP echo requests.

In tunnel mode the outer headers′ IP addresses does not need to be the same as the inner headers′ addresses. For example two security gateways may operate an AH tunnel which is used to authenticate all traffic between the networks they connect together. This is a very typical mode of operation. 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.

Note: The original AH specification in RFC 1825 only mentions tunnel mode in passing, not as a requirement. Because of this, there are IPSec implementations based on that RFC that do not support AH in tunnel mode. This has implications in the ability to implement certain scenarios, such as the one described in Chapter 7, “Remote Access Scenario” on page 133.

3.2.3 IPv6 Considerations

AH is an integral part of IPv6. In an IPv6 environment, 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. Figure 22 on page 46 illustrates the positioning of the AH header in transport mode for a typical IPv6 packet. The position of the extension headers marked with * is variable, if present at all.

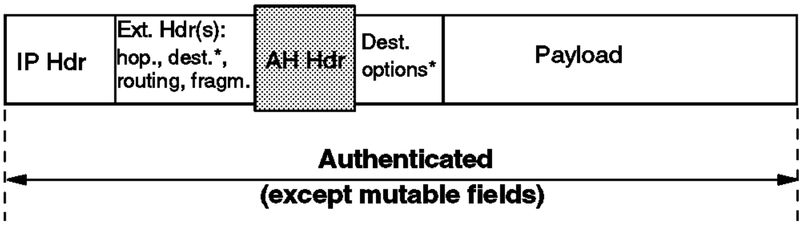


Figure 22. AH in Transport Mode for IPv6

For a detailed description of AH in IPv6 please refer to the current Internet Draft.

3.3 Encapsulating Security Payload (ESP)

ESP is used to provide integrity check, authentication and encryption to IP datagrams. Optional replay protection is also possible. These services are connectionless, they operate on a per-packet basis. The set of desired services are selectable upon SA establishment. 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 the other services. It is highly recommended that if encryption is enabled, then integrity check and authentication be turned on. If only encryption is used, intruders could forge packets in order to mount cryptanalytic attacks. This is infeasible when integrity check and authentication are in place.

Although both authentication (with integrity check) and encryption are optional, at least one of them is always selected. Otherwise it really does not make sense to use ESP at all.

ESP is identified by protocol number 50, assigned by the IANA. 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 processing is applied only to non-fragmented IP packets. However an IP packet with ESP applied can be fragmented by intermediate routers. In this case the destination first reassembles the packet and then applies ESP processing to it. If an IP packet that appears to be a fragment (offset field is non-zero, or the More Fragments bit is set) is input to ESP processing, it is discarded. This prevents the overlapping fragment attack mentioned in 3.2, “Authentication Header (AH)” on page 42.

If both encryption and authentication with integrity check are selected, then the receiver first authenticates the packet and only if this step was successful proceeds with decryption. This mode of operation saves computing resources and reduces the vulnerability to denial of service attacks.

3.3.1 ESP Packet Format

The current ESP packet format is described in the Internet Draft *draft-ietf-ipsec-esp-v2-05.txt*, dated March 1998. It contains important modifications compared to the previous ESP specification, RFC 1827. The information in this section is based on the respective Internet Draft.

The format of the ESP packet is more complicated than that of the AH packet. Actually there is not only an ESP header, but also an ESP trailer and ESP authentication data (see Figure 23). The payload is located (*encapsulated*) between the header and the trailer, hence the name of the the protocol.

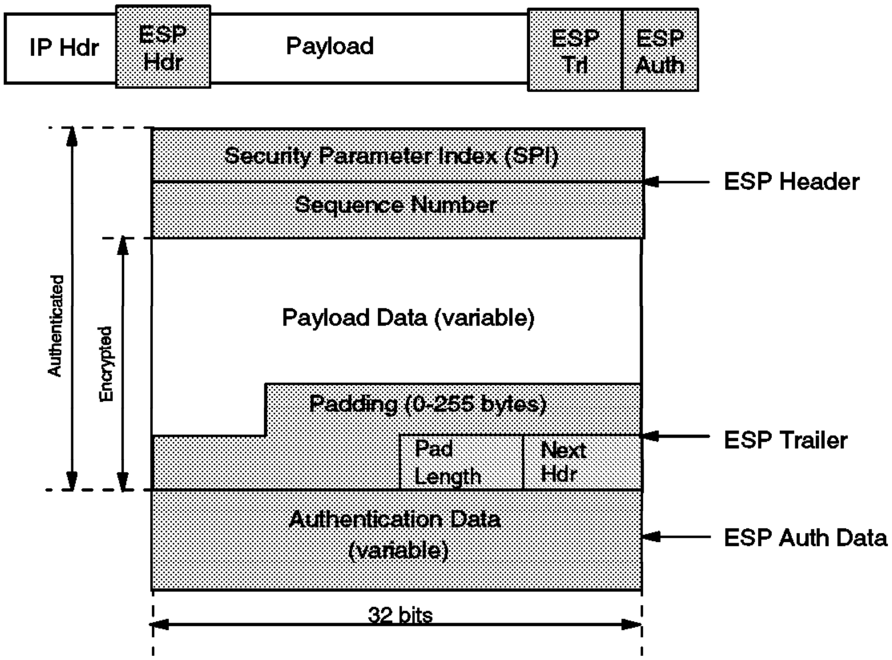


Figure 23. ESP Header and Trailer

The following fields are part of an ESP packet:

Security Parameter Index (SPI)

This field is 32 bits in length. See Security Parameter Index (SPI) on page 39 for the definition.

Sequence Number

This 32-bit field is a monotonically increasing counter. See Sequence Number on page 43 for the definition.

Notes:

1. Typically the anti-replay mechanism is not used with manual key management.
2. The original ESP specification in RFC 1827 did not discuss the concept of sequence numbers. Older IPSec implementations that are based on that RFC can therefore not provide replay protection.

Payload Data

The Payload Data field is mandatory. It consists of a variable number of bytes of data described by the Next Header field. This field is encrypted with the cryptographic algorithm selected during SA establishment. If the algorithm requires initialization vectors, these are also included here.

The ESP specification require support for the DES algorithm in CBC mode (DES-CBC transform). Often other encryption algorithms are also supported, such as triple-DES and CDMF in the case of IBM products.

Padding

Most encryption algorithms require that the input data must be an integral number of blocks. Also, the resulting ciphertext (including the Padding, Pad Length and Next Header fields) must terminate on a 4-byte boundary, so that Next Header field is right aligned. That′s why this variable length field is included. It can be used to hide the length of the original messages too. However, this could adversely impact the effective bandwidth. Padding is an optional field.

Note: The encryption covers the Payload Data, Padding, Pad Length and Next Header fields.

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, for example an upper-level protocol identifier such as TCP. The values are chosen from the set of IP Protocol Numbers defined by the IANA.

Authentication Data

This field is variable in length and contains the ICV calculated for the ESP packet from the SPI to the Next Header field inclusive. The Authentication Data field is optional. It is included only when integrity check and authentication have been selected at SA initialization time.

The ESP specifications require two authentication algorithms to be supported: HMAC with MD5 and HMAC with SHA-1. Often the simpler keyed versions are also supported by the IPSec implementations.

Notes:

1. The IP header is not covered by the ICV.
2. The original ESP specification in RFC 1827 discusses the concept of authentication within ESP in conjunction with the encryption transform. That is, there is no Authentication Data field and it is left to the encryption transforms to eventually provide authentication.

3.3.2 Ways of Using ESP

Like AH, ESP can be used in two ways: transport mode and tunnel mode.

3.3.2.1 ESP in Transport Mode

In this mode the original IP datagram is taken and the ESP header is inserted right after the IP header, as it is shown in Figure 24 on page 49. 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.

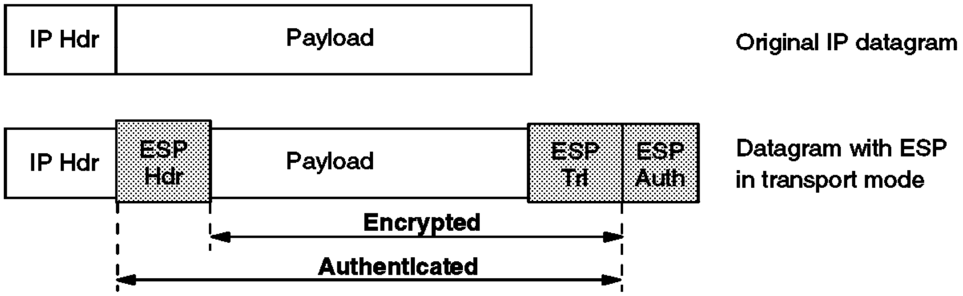


Figure 24. ESP in Transport Mode

ESP in transport mode provides neither authentication nor encryption for the IP header. This is a disadvantage, since false packets might be delivered for ESP processing. The advantage of transport mode the lower processing overhead.

As in the case of AH, ESP in transport mode is used by hosts, not gateways. Gateways are not even required to support transport mode.

3.3.2.2 ESP in Tunnel Mode

As expected, this mode applies the tunneling principle. A new IP packet is constructed with a new IP header and then ESP in transport mode is applied, as illustrated in Figure 25. Since the original datagram becomes the payload data for the new ESP packet, its protection is total if both encryption and authentication are selected. However, the new IP header is still not protected.

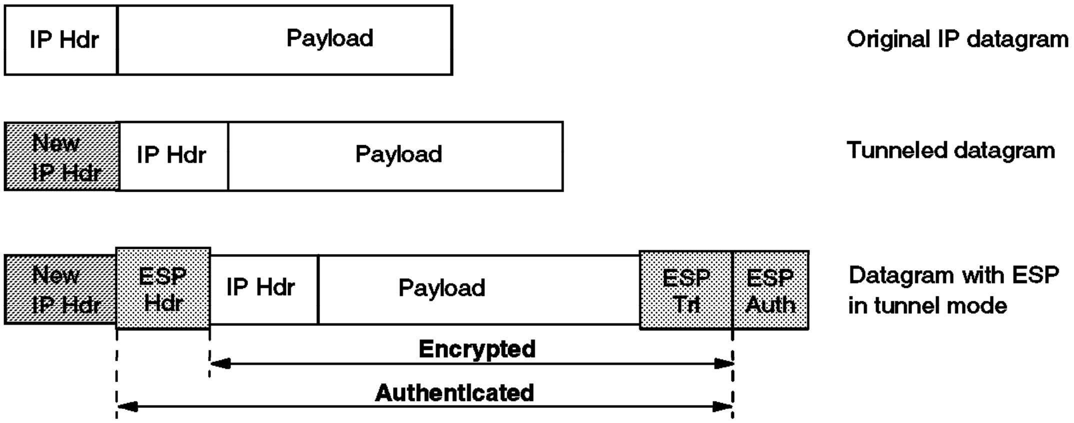


Figure 25. ESP in Tunnel Mode

The tunnel mode is used whenever either end of a security association is a gateway. Thus, between two firewalls the tunnel mode is always used.

Although gateways are supposed to support tunnel mode only, often they can also work in transport mode. This mode is allowed when the gateway acts as a host, that is in cases when traffic is destined to itself. Examples are SNMP commands or ICMP echo requests.

In tunnel mode the outer headers′ IP addresses does not need to be the same as the inner headers′ addresses. For example two security gateways may operate an ESP tunnel which is used to secure 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.

3.3.3 IPv6 Considerations

Like AH, ESP is an integral part of IPv6. In an IPv6 environment, ESP 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. Figure 26 illustrates the positioning of the AH header in transport mode for a typical IPv6 packet. The position of the extension headers marked with * is variable, if present at all.

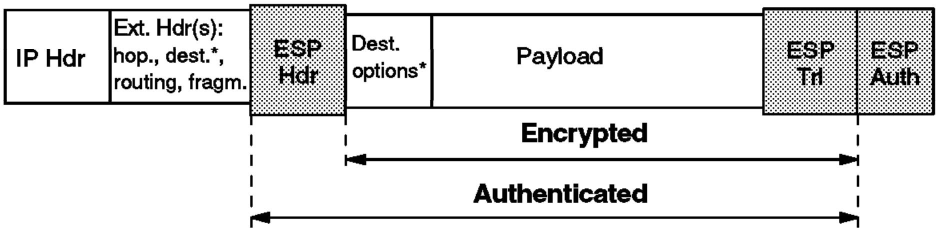


Figure 26. ESP in Transport Mode for IPv6

For more details, please refer to the respective Internet Draft.

3.3.4 Why Two Authentication Protocols?

Knowing about the security services of ESP, one might ask if there is really a requirement for AH. Why does ESP authentication not cover the IP header as well? There is no official answer to these questions, but here are some points that justify the existence of two different IPSec authentication protocols:

- ESP requires strong cryptographic algorithms to be implemented, whether it will actually be used or not. Strong cryptography is an over-hyped and sensitive topic in some countries, with restrictive regulations in place. It might be troublesome to deploy ESP-based solutions in such areas. However, authentication is not regulated and AH can be used freely around the world.
- Often only authentication is needed. While ESP could have been specified to cover the IP header as well, AH is more performant compared to ESP with authentication only, because of the simpler format and lower processing overhead. It makes sense to use AH in these cases.
- Having two different protocols means finer-grade control over an IPSec network and more flexible security options. By nesting AH and ESP for example, one can implement IPSec tunnels that combine the strengths of both protocols.

3.4 Combining IPSec Protocols

The AH and ESP protocols can be applied alone or in combination. Given the two modes of each protocol, there is quite a number of possible combinations. To make things even worse, the AH and ESP SAs do not need to have identical endpoints, so the picture becomes rather complicated. Luckily, out of the many possibilities only a few make sense in real-world scenarios.

Note: The *draft-ietf-ipsec-arch-sec-04.txt* Internet Draft is the current document that describes the mandatory combinations that must be supported by each IPSec implementation. Other combinations may also be supported, but this might impact interoperability.

We mentioned in 3.1.1, “Security Associations” on page 39 that the combinations of IPSec protocols are realized with SA bundles.

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, more than three levels are inpractical.

These approaches can be combined, for example an IP packet with transport adjacency IPSec headers can be sent through nested tunnels.

When designing a VPN, one should limit the IPSec processing stages applied to a certain packet to a reasonable level. In our view three applications is that limit over which further processing has no benefits. Two stages are sufficient for almost all the cases.

Note that in order to be able to create an SA bundle in which the SAs have different endpoints, at least one level of tunneling must be applied. Transport adjacency does not allow for multiple source/destination addresses, because only one IP header is present.

The practical principle of the combined usage is that upon the receipt of a packet with both protocol headers, the IPSec processing sequence should be authentication followed by decryption. It is a common sense decision not to bother with the decryption of packets of uncertain origin.

Following the above principle, the sender first applies 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, a new question arises: should ESP authentication be turned on? AH authenticates the packet anyway. The answer is simple. Turning ESP authentication on makes sense only when the ESP SA extends beyond the AH SA, as in the case of the supplier scenario. In this case, not only does it make sense to use ESP authentication, but it is highly recommended to do so, to avoid spoofing attacks in the intranet.

As far as the modes are concerned, the usual way is that 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.

Let′s take a systematic look on the plausible ways of using the IPSec protocols, from the simplest to the more complicated nested setups. You learn the details on how these cases are applied to real life scenarios in Part 3, “VPN Scenarios and Implementation” on page 79.

3.4.1 Case 1: End-to-End Security

As it is shown in Figure 27, two hosts are connected through the Internet (or an intranet) without any IPSec gateway between them. They can use ESP, AH or both. Either transport or tunnel mode can be applied.

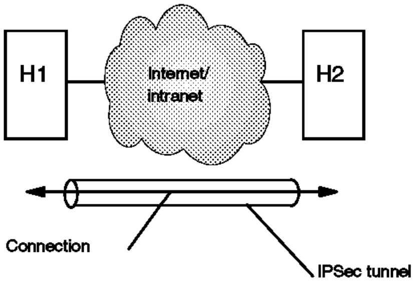


Figure 27. End-to-End Security

The combinations required to be supported by any IPSec implementation are the following:

Transport Mode

1. AH alone
2. ESP alone
3. AH applied after ESP (transport adjacency)

Tunnel Mode

1. AH alone
2. ESP alone

3.4.2 Case 2: Basic VPN Support

Figure 28 illustrates the simplest VPN. The gateways G1 and G2 run the IPSec protocol stack. The hosts in the intranets are not required to support IPSec.

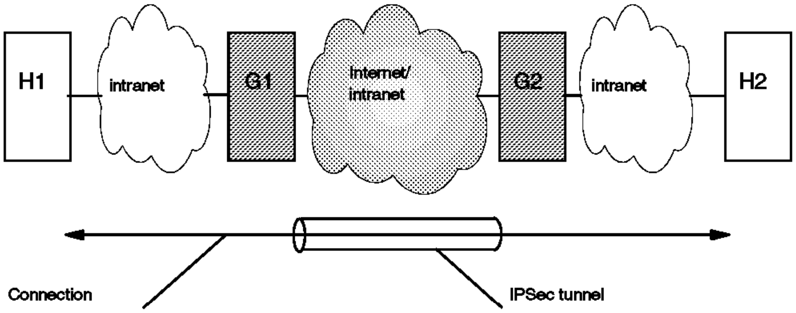


Figure 28. Basic VPN Support

In this case the gateways are required to support only tunnel mode, either with AH or ESP.

3.4.2.1 Combined Tunnels between Gateways

Although the gateways are required to support only an AH tunnel or ESP tunnel, often it is desirable to have tunnels between gateways that combine the features of both IPSec protocols.

The IBM IPSec implementations support this type of combined AH-ESP tunnels. The order of the headers is user selectable by setting the tunnel policy. (See 4.1.1.2, "Policies" on page 59 for more details.)

A combined tunnel between the gateways does not mean that iterated tunneling takes place. Since the SA bundle comprising the tunnel have identical endpoints, it is inefficient to do iterated tunneling. Instead, one IPSec protocol is applied in tunnel mode and the other in transport mode, which can be conceptually thought of as a combined AH-ESP tunnel. An equivalent approach is to IP tunnel the original datagram and then apply transport adjacency IPSec processing to it. The result is that we have an outer IP header followed by the IPSec headers in the order set by the tunnel policy, then the original IP packet, as it is shown in the figure below. This is the packet format in a combined AH-ESP tunnel between two IBM firewalls.

Note: The ESP authentication data is not present because the IPSec implementation in the IBM firewall does not support the new specifications yet.

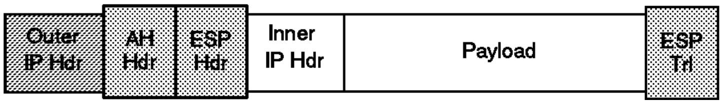


Figure 29. Combined AH-ESP Tunnel

3.4.3 Case 3: End-to-End Security with VPN Support

This case is a combination of cases 1 and 2 and it does not raise new IPSec requirements for the machines involved (see Figure 30). The big difference from case 2 is that now the hosts are also required to support IPSec.

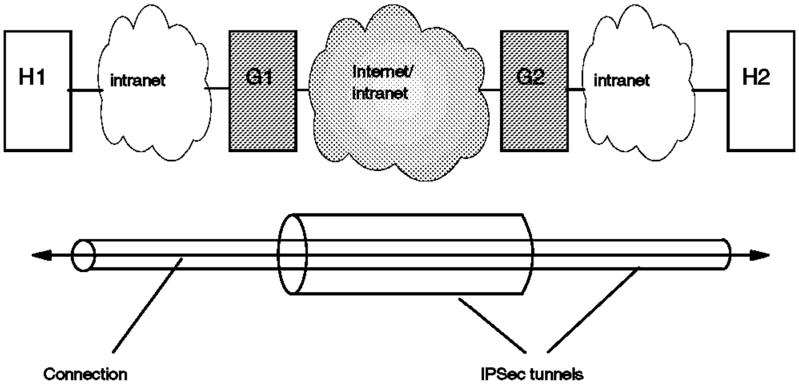


Figure 30. End-to-End Security with VPN Support

In a typical setup, the gateways use AH in tunnel mode, while the hosts use ESP in transport mode. An enhanced security version could use a combined AH-ESP tunnel between the gateways. In this way the ultimate destination addresses would be encrypted, the whole packet traveling the Internet would be authenticated and the carried data double encrypted. This is the only case when three stages of IPSec processing might be useful, however, at a cost; the performance impact is considerable.

3.4.4 Case 4: Remote Access

This case, shown in Figure 31, applies to the remote hosts that use the Internet to reach a server in the organization protected by a firewall. The remote host commonly uses a PPP dial-in connection to an ISP.

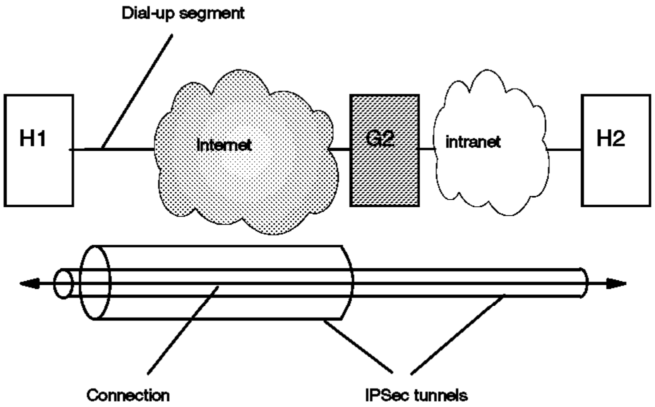


Figure 31. Remote Access

Between the remote host H1 and the firewall G2 only tunnel mode is required. The choices are the same as in case 2. Between the hosts themselves either tunnel mode or transport mode can be used, with the same choices as in case 1.

A typical setup is to use AH in tunnel mode between H1 and G2 and ESP in transport mode between H1 and H2. Older IPSec implementations that do not support AH in tunnel mode cannot implement this.

It is also common to create a combined AH-ESP tunnel between the remote host H1 and the gateway G2. In this case H1 can access the whole intranet with using just one SA bundle, whereas if it were using the setup shown in Figure 31, it only could access one host with one SA bundle.

3.4.5 Conclusion and an Example

While the combination of the IPSec protocols in theory leads to a large number of possibilities, in practice only a few (those presented above) are used. One very common combination is AH in tunnel mode protecting ESP traffic in transport mode. Combined AH-ESP tunnels between firewalls are also frequent.

Figure 32 on page 55 shows in detail how the first combination is realized. Consider that host H1 in Figure 30 on page 53 sends an IP packet to host H2. Here is what happens:

1. Host H1 constructs the IP packet and applies ESP transport to it. H1 then sends the datagram to gateway G1, the destination address being H2.
2. Gateway G1 realizes that this packet should be routed to G2. Upon consulting its IPSec databases (SPD and SAD) G1 concludes that AH in tunnel mode must be applied before sending the packet out. It does the required encapsulation. Now the IP packet has the address of G2 as its destination, the ultimate destination H2 being encapsulated.
3. Gateway G2 receives the AH-tunneled packet. It is destined to itself, so it authenticates the datagram and strips off the outer header. G2 sees that the payload is yet another IP packet (that one sent by H1) with destination H2, so it forwards to H2. G2 does not care that this packet has an ESP header.

4. Finally H2 receives the packet. As this is the destination, ESP-transport processing is applied and the original payload retrieved.

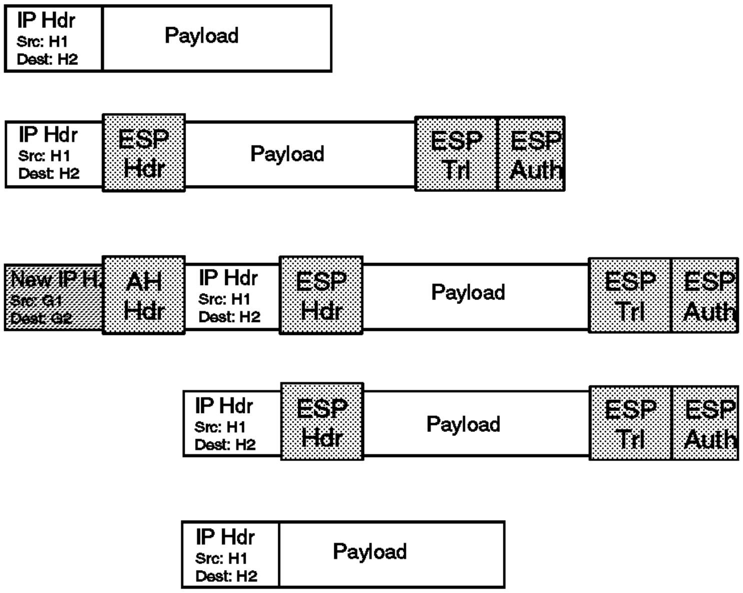


Figure 32. Nesting of IPSec Protocols

3.5 Current IPSec Internet Drafts

By the time this book was published, the current Internet Draft specifications for the core components of IPSec were the following:

IP Security Architecture (IPSec)
    http://www.ietf.org/internet-drafts/draft-ietf-ipsec-arch-sec-05.txt

IP Authentication Header (AH)
    http://www.ietf.org/internet-drafts/draft-ietf-ipsec-auth-header-06.txt

IP Encapsulating Security Payload (ESP)
    http://www.ietf.org/internet-drafts/draft-ietf-ipsec-esp-v2-05.txt

Internet Security Association and Key Management Protocol (ISAKMP)
    http://www.ietf.org/internet-drafts/draft-ietf-ipsec-isakmp-09.txt

Oakley Key Determination Protocol
    http://www.ietf.org/internet-drafts/draft-ietf-ipsec-oakley-02.txt

Internet Key Exchange (IKE)
    http://www.ietf.org/internet-drafts/draft-ietf-ipsec-isakmp-oakley-07.txt

IPSec Domain of Interpretation for ISAKMP
    http://www.ietf.org/internet-drafts/draft-ietf-ipsec-ipsec-doi-09.txt

Part 2. IBM eNetwork VPN Solutions

Chapter 4. IBM eNetwork VPN Solutions

The description of the IBM eNetwork VPN solutions in the following sections lists the implementation of VPN features on the various IBM platforms and positions the different products. Towards the end of the chapter there are two tables. One covers the interoperability between IBM products, the other shows how IBM solutions interoperate with other vendors′ VPN products.

4.1 Key Terms and Features

As discussed in Chapter 1, “Virtual Private Networks (VPN) Overview” on page 3 all IBM eNetwork VPN solutions are based on IPSec, because it is the only available framework addressing most of the aforementioned security exposures. Before describing the functions available on the different IBM platforms we need to discuss features and terms commonly used on all platforms.

Note: The IBM Nways router platforms (2210 and 2216) offer layer 2 tunneling functionality (L2TP) and multiprotocol support in addition to IPSec. All those features are described in a separate redbook, *A Comprehensive Guide to Virtual Private Networks, Volume II: IBM Nways Router Solutions*, SG24-5234, to be published later this year.

4.1.1 IPSec Functionality

In this section we discuss all variations found in the IBM eNetwork VPN solutions. However, some products may not offer all options listed. Please refer to Table 1 on page 65 and Table 2 on page 71 for details.

4.1.1.1 IPSec Protocols

The valid security protocols are ESP (RFCs 1827 and 1829) and AH (RFC 1826). Authentication and encryption can be used independently. There are implementations (such as AIX V4.3 and the 2210/2216 Router) that also use headers defined by newer Internet Drafts. (See Chapter 3, “Description of IPSec” on page 39 for an explanation of the differences between the original and the new IPSec headers.)

4.1.1.2 Policies

Your tunnel defined policy specifies that the data (original IP packets) be either:

- Encrypted (encr only)
- Authenticated (auth only)
- Encrypted and then authenticated (encr/auth)
- Authenticated and then encrypted (auth/encr)
- Neither encrypted nor authenticated (none)

The tunnel policy or level of protection depends on your security requirements. A typical scenario may have multiple secure networks (for example, branches of a company that are in different cities) with tunnels between them in order to protect the information. There may be more than one tunnel between a single pair of nodes, which might be useful for different encryption and authentication choices. An example of such a requirement for multiple SAs between nodes is described in Chapter 6, “Business Partner/Supplier Network Scenario” on page 111.

If you use both authentication and encryption we recommend that you use the option encrypted and then authenticated. It will authenticate the full IP datagram, including the encapsulated IP header. The RFCs determine this option as mandatory whereas they do not specify authenticated and then encrypted as a required option. That makes sense because this option would not authenticate the encapsulated IP header.

For more details see Chapter 3, “Description of IPSec” on page 39, and especially 3.4, “Combining IPSec Protocols” on page 50 which outlines common scenarios. The actual implementation details and configurations are included in Part 3, “VPN Scenarios and Implementation” on page 79.

4.1.1.3 Cryptographic Algorithms (Transforms)

Currently the following transforms are used for authentication within the IBM eNetwork VPN solutions:

- KEYED MD5
- HMAC MD5 (with optional replay protection)
- HMAC SHA (with optional replay protection)

The following algorithms are used for encryption/decryption:

- DES CBC 4 (with a 32-bit initialization vector)
- DES CBC 8 (with a 64-bit initialization vector)
- 3DES CBC
- CDMF

Note: With the initial RFCs two packet headers were necessary to provide both authentication and encryption. The new header format drafts offer an ESP header that allows an additional authentication algorithm of your choice (as long it is a transform that uses the new header format). Currently this kind of ESP header can be used in the AIX V4.3 implementation.

4.1.2 Tunnel Types

An IPSec tunnel is defined by specifying a pair of security associations (SAs) between two hosts. A security association is uniquely identified by a triple consisting of a security parameter index (SPI), an IP destination address, and a security protocol (AH or ESP) identifier. The security parameter index (SPI) enables the receiving system to select the security association under which a received packet will be processed. Other elements of the security association (SA) such as the cryptographic algorithms to use, keys and lifetime can be specified, or defaults can be used.

There are two SA types: tunnel mode and transport mode. According to the RFCs firewalls have to use tunnel mode if they are acting as gateways; the implementation of transport mode is optional for firewalls. See Chapter 3, “Description of IPSec” on page 39 for details on IPSec protocols and security associations.

Currently there are three kinds of IPSec tunnels that IBM uses for different purposes in its eNetwork VPN products:

4.1.2.1 Manual Tunnel

A manual tunnel implements the IPSec standards and is typically used between an IBM firewall and a non-IBM firewall. In theory you should be able to use it to connect to any product supporting the IPSec standards. In practice it depends mainly on whether you are able to find a combination of tunnel characteristics (such as transforms, policy and header format) supported by both products. Many vendors offer the transforms keyed MD5 with DES or HMAC MD5 with DES. This is a base subset that works with most implementations of the IPSec RFCs. In our tests we used manual tunnels for instance between an AIX V4.3 server and a Windows 95 system running the eNetwork Communications Suite. (See Chapter 6, “Business Partner/Supplier Network Scenario” on page 111 for the configuration details.)

This tunnel type usually requires all fields to be filled in manually. (Although some products, such as the eNetwork Firewall for AIX, will assign an autogenerated value to some fields; for details see the corresponding product section later in this chapter.)

- Source and destination IP address of the tunnel.
- SA Type: Tunnel or transport mode. If your system acts as a gateway, it has to use tunnel mode. If acting as a host, we recommend that you choose transport mode. It involves less overhead than tunnel mode.

  Note: If you have no such field on a firewall acting as a security gateway, according to the RFCs it has to use tunnel mode. On a client if there is no choice, you may assume that it probably uses transport mode.

- IPSec protocol, policy and authentication/encryption transform: See 4.1.1, “IPSec Functionality” on page 59. The scenarios implemented in Part 3, “VPN Scenarios and Implementation” on page 79 contain examples for different situations.
- Source and destination key: The keys have to match on both tunnel sides. The value your partner has chosen for the source key is your destination key and vice versa.
- Source and destination SPI: There is one SPI pair for AH and another one for ESP (if you use both protocols). The above rule for the keys goes also for the SPIs; the value your partner specified as source SPI is your destination SPI and vice versa. SPI value 0 is reserved to indicate that no security association exists. The set of SPI values in the range of 1 through 255 are reserved by the Internet Assigned Numbers Authority (IANA) for future use.
- Session key lifetime: Specifies the time in minutes that a manual tunnel will be operational. When the tunnel life time is reached, the tunnel will cease operation until you start it again. Note that after only starting the tunnel again the keys remain the same; for security reasons you should establish new keys. Otherwise the chance for a hacker to crack the key gets bigger as he or she has more time to mount new attacks. The value specified will also affect performance (smaller value, bigger performance hit). Generally, values used for CDMF will be smaller than those used for DES due to the strength of the encryption algorithms.
- Tunnel ID: This value has to uniquely identify the tunnel within the tunnel definitions on your system and must match the ID used in the corresponding tunnel definition of your partner. It is actually not required for the security association but for the packet filter rules. If the tunnel ID does not equal zero, the data packet is sent to the IPSec kernel. This field is not required

by the RFCs, but used within all IBM server/gateway products described in this book.

- Replay Prevention: This feature is only available with the new draft header format used for instance by HMAC MD5 and HMAC SHA. Some newer implementations already support replay prevention. For instance, in the AIX V4.3 manual tunnel definition you will find the option of whether you want to use replay prevention or not. If this field exists, make sure that you match your partners definitions or capabilities. If you use keyed MD5, replay prevention is not a valid option.

  For manual tunnels the RFCs recommend not to use replay prevention. The reason is that the RFCs do not allow a wrap arround of the counter that is used for replay prevention. Therefore, if the counter reaches its maximum value the tunnel has to cease and must be started again. With a long session key lifetime this situation is likely. (Please see Chapter 3, “Description of IPSec” on page 39 for details.)

The name manual tunnel already implies that there is no automatic key management. As currently there is no key management protocol specified in the IPSec RFCs IBM will include the ISAKMP/Oakley key management protocol later this year when the corresponding RFCs are in place. Until then as long as you use manual tunnels you have to either refresh the keys manually or inhibit key refreshes at all. (Please see 6.6, “Manual Key Distribution” on page 131 for considerations on this topic.) Sometimes this manual key distribution is also referred to as *preshared keys*. Because of the administration overhead currently involved with manual tunnels for the short term we recommend you use IBM tunnels whenever possible. They offer a solution to the key refresh issue.

4.1.2.2 IBM Tunnel

This kind of tunnel uses IP Security Protocol (IPSP) which is an IBM unique protocol. It features an automatic key update mechanism, using UDP port 4001. Under this scheme, a new encryption key is generated at regular intervals and communicated through the tunnel encrypted under the current key. With IBM tunnels there is no need (and in fact also no option) to specify the SPIs and the keys; they are automatically determined by the software. There is also no choice for the authentication algorithm; IBM tunnels always use keyed MD5. On the other hand you have to specify options not found in a manual tunnel definition.

Besides some of the fields discussed in the manual tunnel section above (tunnel ID, source and destination IP address, policy and ESP transform), you will find the following additional fields when defining an IBM tunnel:

- Initiator: Determines who starts the session negotiation. If you are not sure what your partner has specified, set it to yes. If both partners specify yes, the tunnel logic will resolve the deadlock. If no partner sets it to yes, the tunnel will not operate at all.
- Session Key Lifetime: This field is very similar to the one described in the section on manual tunnels. It specifies the time in minutes where the current session key may be used. The big difference is, however, that a new key will be automatically created before the old key expires. Therefore, the IBM tunnel does not cease operation after the key lifetime has elapsed.
- Session Key Refresh Time: Specifies the time in minutes between a new key start and an old key expiration. If for instance the refresh time is 10 minutes, then the old and the new key are both valid during the last 10 minutes of the

session key lifetime. Therefore the value must be equal or less than the session key lifetime. The key refresh time should be half of the session key lifetime.

As the name implies already this tunnel type was developed for use with IBM products. Currently it allows, for instance, to establish a connection between two AIX systems (could be AIX firewalls and/or AIX V4.3 systems).

We used the IBM tunnel extensively between AIX firewalls. It is convenient for administrators because they don′t need to worry about the key refresh. See Chapter 5, "Branch Office Connection Scenario" on page 81 for configuration details.

4.1.2.3 Dynamic Tunnel

A dynamic tunnel is a special variation of a manual tunnel and is an implementation only found on the IBM firewall. It also uses the IPSec standards, but there are only two IBM clients it is used with:

- Windows 95 IPSec Client (supplied with the eNetwork Firewall for AIX)
- OS/2 TCP/IP V4.1 IPSec Client (part of the OS/2 TCP/IP V4.1 stack)

The reason for calling the tunnel dynamic is that the tunnel definition is not based on the client′s IP address but on a client target user. Therefore the client′s IP address does not have to be known. This is important because a remote client usually uses a dynamic IP address supplied by the provider when connecting through the Internet to the firewall.

The connection is established by using the Secure Sockets Layer (SSL) protocol. The firewall destination port is 4005. The dynamic tunnel on the firewall is not activated until the client starts the tunnel. When the user selects **Connect tunnel**, the client starts an SSL control session with the firewall. The SSL server application authenticates the remote client based on the (already encrypted) user ID and password and sends the tunnel policy to the client. Because the client IP address is not known by the time of the tunnel definition the necessary firewall filter rules to connect to the secure network are also generated dynamically by the firewall when the tunnel is started by the client.

Note: The dynamic filter rules are put at the top of all filter rules and hence are evaluated first. This also means that on the firewall you have no possibility (as you have with the other tunnel types) to further restrict the traffic between the client and the secure network.

After the tunnel startup the SSL control session is terminated and the user is now able to work with systems on the secure network. The dynamic filters remain active until the user at the client disconnects the tunnel or the tunnel times out.

When the user selects **Disconnect tunnel**, the client starts a new SSL control session with the firewall. The SSL server application authenticates the client again based on user ID and password and deactivates the dynamic tunnel, dynamic filters, and dynamic policy for the remote user. Then the SSL control session is terminated.

The only special field in the definition of a dynamic tunnel is the Target User field. As discussed above it replaces the destination IP address found in the manual tunnel. All other fields were already described in 4.1.2.1, "Manual Tunnel" on page 61. See Chapter 7, "Remote Access Scenario" on page 133 for configuration details.

Now that you know what tunnel types IBM uses in its products we have a look at some other functions related to VPNs.

Notes:

1. Keep in mind that IBM′s manual tunnel is the one that corresponds to the RFCs. Therefore this is the tunnel type to use if your partner uses a vendor product. The other two tunnel types are solely for use with IBM products.

2. IBM has announced that it will offer standard-compliant automatic key exchange and refresh capabilities for its eNetwork VPN solutions later this year.

4.1.3 Other Important Features

In order to build your VPN solution some other features may be required or helpful:

Packet Filtering
Packet filtering will help you to restrict VPN traffic to certain systems in your own or your suppliers/business partners network. It is helpful if packet filters are used together with logging.

Logging
Logging allows you to learn more about what happened to the data packets and about the current VPN status.

IPv6 Support
For the future it will also be important to know whether a product supports IPv6.

Modular Support for New Cryptographic Transforms
As the cryptographic algorithms are under continuous improvement it is desirable that the design of the VPN products allows plug-in and replaceable kernel modules for encryption and authentication.

We have included the above features in our evaluations. You will find the results in the specific product sections.

4.2 Server/Gateway Platforms

Based on typical usage we decided to put the following products in this category:

- eNetwork Firewall for AIX V3
- AIX V4.3
- 2210 Nways Multiprotocol Router and 2216 Nways Multiaccess Connector
- OS/390 Server
- 3746 Multiaccess Enclosure

Note: On the Windows NT and AS/400 platforms the IBM firewall products do not yet implement VPN/IPSec functionality but will do so later this year.

The following table reflects the features discussed at the beginning of this chapter. In addition we included other general VPN related product information in the product sections. For detailed configuration steps see Part 3, “VPN Scenarios and Implementation” on page 79. The interoperability test section later in this chapter lists configuration options we recommend in order to set up a connection between two products.

Table 1. Server Platforms - Supported VPN Features

Feature		eNetwork Firewall for AIX	AIX V4.3	2210 / 2216 / 3746	OS/390 Server
Tunnel Type	IBM	√	√		
	Manual	√	√	√	√
	Dynamic	√			
IPSec Protocol	AH	√	√	√	√
	ESP	√	√	√	√
Header Format	RFCs 18xx	√	√		√
	New Draft Headers		√	√	
SA Type	Transport Mode		√	√	
	Tunnel Mode	√	√	√	√
AH Transform	Keyed MD5	√	√		√
	HMAC MD5		√	√	
	HMAC SHA		√	√	
ESP Transform	DES CBC 4	√	√		√
	DES CBC 8	√	√	√	√
	CDMF	√	√	√	√
	3DES CBC		√ (V4.3.1)	√	
	Authentication		√	√ (optional)	
Other	Packet Filters	√	√	√	√
	Logging	√	√	√ (remote)	√
	Plug-ins		√		
	IP Version 6		√		

4.2.1 IBM eNetwork Firewall for AIX

To the best of our knowledge the eNetwork Firewall for AIX was the first IBM product to offer an IPSec implementation (in SNG V.2.1, which was available in October 1995).

We have used the eNetwork Firewall for AIX in several scenarios. See Chapter 5, “Branch Office Connection Scenario” on page 81 for details on how to configure an IBM tunnel between two AIX firewalls. For information on how to set up the eNetwork Firewall for AIX in general we recommend the redbook *Protect and Survive Using IBM Firewall 3.1 for AIX*, SG24-2577.

Currently the AIX V4.3 operating system offers richer VPN functionality than the eNetwork Firewall for AIX. Therefore the IPSec functionality of AIX V4.3.1 will be included in the eNetwork Firewall for AIX later this year.

4.2.1.1 Product Versions

Since the last program services for the Secured Network Gateway (SNG) V2.2 will end on June 24, 1998, we did not include this version in our tests. Concerning VPN features (such as protocols, transforms and policies) SNG V2.2 is identical to the functionality found in AIX firewall V3.1. If you still use SNG V2.2, please make sure that you have applied the latest available service level.

From a functional VPN point of view the eNetwork Firewall for AIX V.3.2 is almost identical to V.3.1. The Win95 IPSec client has changed as discussed in 4.3.2.3, “eNetwork Firewall for AIX V3.2” on page 73, reflecting enhancements to the remote user authentication methods.

In eNetwork Firewall for AIX V3.1 some VPN-related defects have been discovered. At the time of writing this redbook all known VPN defects have been fixed in V3.1.1.5 (which was the current code level at that time) and we had no problems after we upgraded our machines to this level of code. Therefore we strongly recommend you upgrade your firewall to the latest available level if you plan to use VPN features. We also used eNetwork Firewall for AIX V.3.2.1 in some scenarios.

4.2.1.2 VPN Features

The eNetwork Firewall for AIX currently offers:

- Tunnel types: All (IBM, manual and dynamic)
- SA Type: Tunnel Mode

  There is no option to choose between tunnel or transport mode; the firewall will always automatically use tunnel mode.
- IPSec protocols: All (AH and ESP)
- Header Formats: RFCs 18xx
- Policies: All (auth, encr, encr/auth, auth/encr)
- AH transform: Keyed MD5
- ESP transforms: DES CBC 4, DES CBC 8 and CDMF
- Other VPN-related features: Packet filtering and logging

4.2.1.3 Hints on Options and Manual Tunnels

The eNetwork Firewall for AIX currently does not support the following options:

- Replay prevention
- Tunnel life time of 0 (unlimited)

This is important to know if your tunnel partner offers this options, such as an AIX V4.3 system which offers both options.

The eNetwork Firewall for AIX creates the values for the SPIs and the keys automatically and puts them in the export files of the tunnel definition. If your partner product has no option for importing a tunnel definition from the firewall, you still need to export the tunnel definitions and have a look at the key and SPI values of the corresponding export files. This will enable you to match them in your partner product′s tunnel definition.

4.2.2 AIX V4.3

Starting with AIX V4.3 the operating system offers a rich set of VPN features. With its included packet filtering and logging functionality it could even be used as an entry firewall.

If you are on an AIX V4.3.0 system we recommend you upgrade to AIX V4.3.1. It offers additional VPN functions. See 4.2.2.6, "AIX V4.3.1 - New Functions" on page 69 for details on what exactly has changed.

For details on what has changed between AIX V4.2 and AIX V4.3 we recommend the redbook *AIX Version 4.3 Differences Guide*, SG24-2014.

4.2.2.1 VPN Features

AIX V4.3.1 offers:

- Tunnel types: IBM and manual
- SA Types: All (tunnel and transport mode)
- IPSec protocols: All (AH and ESP)
- Header Formats: All (RFCs 18xx and new draft headers)
- Policies: All (auth, encr, encr/auth, auth/encr)
- AH transforms: All (Keyed MD5, HMAC MD5 and HMAC SHA)
- ESP transforms: All (DES CBC 4, DES CBC 8, CDMF, 3DES CBC and the new ESP authentication)

  With the initial RFCs two packet headers were necessary to provide both authentication and encryption. The new ESP header format drafts offer an ESP header that allows an additional authentication algorithm of your choice (as long as it is a transform that uses the new header format).

- Other VPN-related features: All (packet filtering, logging, plug-ins and IPv6)

4.2.2.2 Running the Firewall Software on AIX V4.3

AIX V4.3 IPSec interoperates with the eNetwork Firewall for AIX. Although both the firewall and AIX V4.3 support IPv4, firewall code supersedes the AIX IPSec code in providing IPv4 secure tunnel support. This means that if a firewall is configured on a V4.3 system, IPv4 will use the firewall IPSec code. Note that files belonging to the AIX IPSec fileset are unique, so both products can be installed without overwriting each other′s files. If AIX IPv4 detects that firewall code is installed, it does not load its own IPSec Version 4 code.

eNetwork Firewall for AIX V3.2 is the first release officially supported on AIX V4.3. Firewall V3.1 does not work on AIX V4.3 and is therefore not supported.

4.2.2.3 Hints on Policies

The SMIT IPSec panels presented in AIX V4.3.0 do not offer the full functionality found on the command line. The following configurations are valid and worked in our manual tunnel tests, but can only be configured by using the command line (or by changing a tunnel that was created via SMIT, using the chtun command):

- Policy encr/auth (also called ea)
- Policy encr only (also called e) with the old RFC headers

In AIX V4.3.1 the SMIT panels for tunnel creation show the same behavior, but the full functionality is now also offered in the SMIT panels for changing tunnel definitions. Before AIX V4.3.1 you needed to use the command line to achieve this. The reason for this somewhat complicated kind of implementation lies in the rich functionality of AIX V4.3 IP Security, which makes it difficult to have a simple SMIT panel structure without loosing some valid combinations.

4.2.2.4 Host-Firewall-Host Tunnel Option

This is a special option we only found on the AIX V4.3 platform. It allows you to establish a tunnel between your AIX V4.3 local host (H1) and a remote firewall (G2) and automatically creates the filter rules needed on your AIX 4.3 system to connect to your real destination host (H2) or network behind the remote firewall. Figure 33 illustrates the above scenario.

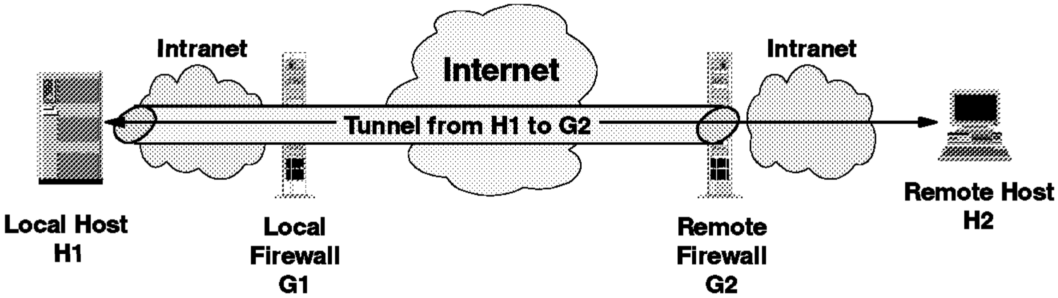


Figure 33. Host-Firewall-Host Tunnel

In our opinion you will rarely need to use this option. It means that you don′t trust your own secure network (therefore establishing one tunnel endpoint on your local host), but you trust your partner′s secure network (therefore making the remote firewall the other endpoint of the tunnel). The traffic within your partners secure network would then be in cleartext. Given the above case you would then probably also trust your own secure network. Therefore the tunnel endpoints would be the firewalls on both sides and not the endpoints of the real traffic. This setup coincides with our branch office scenario described in Chapter 5, “Branch Office Connection Scenario” on page 81.

The host-firewall-host option makes sense if your tunnel partner has an AIX 4.3 server and you do not trust his or her secure network. In this case your tunnel would have your firewall and the remote AIX server as end points. Your partner would then need to use this option for his or her configuration (if he or she trusts your secure network, that is). If you do not trust the secure networks at all, the business partner/supplier scenario described in Chapter 6, “Business Partner/Supplier Network Scenario” on page 111 is the one you will probably use.

From a technical point of view the host-firewall-host option is actually not about the tunnel definition but the filter definitions. For achieving the same functionality as with the host-firewall-host option you could as well use the host-host option and specify the needed filter rules on the advanced SMIT IPSec panels. Therefore it is just an additional option for your convenience; don′t get confused by it.

4.2.2.5 Hints on Manual and IBM Tunnels

AIX V4.3 does not offer a field to enter a tunnel ID. The tunnels are automatically numbered (starting by 1). If your partner uses the tunnel ID field, you have to take care that the auto-generated IDs are not already in use on his or her system.

When entering the keys for manual tunnels, don′t forget to put 0x in front of the key. SMIT expects to get a hexadecimal value.

Session key refresh is not supported for IPv6 tunnels.

4.2.2.6 AIX V4.3.1 - New Functions

The new AIX version offers enhanced VPN functionality:

- New transform: Triple DES encapsulation (included in AIX Version 4.3 Bonus Pack for the U.S.).
- Increased options in pairing of authentication and encapsulation algorithms.

  Note: When using the combined ESP header, AIX V4.3.1 allows any AH/ESP combination as long as it uses the new header format. Only keyed MD5 and DES CBC 4 are not able to use the new header format.
- Performance improvements in both HMAC-SHA and DES for PowerPC-based platforms.
- Crypto extensions can be dynamically loaded and unloaded.
- Improved logging: For instance, authentication violations are logged now.

The AIX V4.3 platform was the second server/gateway platform we used extensively during our tests. Please see Chapter 6, “Business Partner/Supplier Network Scenario” on page 111 for the necessary configuration steps, and see also 9.2, “Additional AIX V4.3 Combinations” on page 174 for further IPSec combinations with AIX V4.3.

4.2.3 IBM Nways Routers

The current common code base for the IPSec functionality is:

- 2210 Router: Multiprotocol Routing Services (MRS) V3.1
- 2216 Router: Nways Multiprotocol Access Services (MAS) V3.1

4.2.3.1 VPN Features

The two router families offer:

- Tunnel type: Manual
- SA Types: All (tunnel and transport mode)
- IPSec protocols: All (AH and ESP)
- Header Formats: New draft headers
- Policies: All (auth, encr, encr/auth, auth/encr)
- AH transforms: HMAC MD5 and HMAC SHA
- ESP transforms: DES CBC 8, CDMF, 3DES, and the new ESP authentication
- Other VPN-related features: Packet filtering and remote logging

Volume II of the VPN redbooks deals with the VPN implementation in IBMs router products. Please see *A Comprehensive Guide to Virtual Private Networks, Volume II: IBM Nways Router Solutions*, SG24-5234, to be published later this year.

4.2.4 IBM 3746 Multiaccess Enclosure

As the 3746 MAE is also common code based, it supports the same IPSec features that the 2210 and 2216 Nways routers support. These functions were announced May 5, 1998 with a planned general availability of October 30, 1998.

4.2.5 OS/390 Server

The IPSec functionality on an OS/390 Server is based on the eNetwork Communications Server for OS/390, V2R5.

4.2.5.1 VPN Features

Currently an OS/390 Server offers:

- Tunnel types: Manual
- SA Types: Tunnel Mode
- IPSec protocols: All (AH and ESP)
- Header Formats: RFCs 18xx
- Policies: All (auth, encr, encr/auth, auth/encr)
- AH transforms: Keyed MD5
- ESP transforms: DES CBC 4, DES CBC 8 and CDMF
- Other VPN-related features: Packet filtering and logging.

Support for transport mode tunnels will be added in the next release of eNetwork Communications Server for OS/390. For further information on OS/390 Server please see the redbook *Stay Cool on OS/390: Installing Firewall Technology*, SG24-2046, and see also 9.3, “OS/390 Server Combinations” on page 174.

4.3 Client Platforms

This section covers the IBM VPN client products. Of course systems described in the server section can also be used as VPN clients. Actually from the VPN point of view there is no server and client, but a tunnel owner and partner. If you compare the following feature table to the server table, you will realize additional check boxes to indicate whether the product has dial-up capabilities and/or LAN connectivity. We named the corresponding rows LAN and Dial-up.

None of the clients in the table below supports the IBM tunnel (as it was developed for firewall-to-firewall VPNs) or the more advanced algorithms (HMAC SHA, DES CBC MD5 and 3DES); therefore we removed those features from the client table.

Note: IBM has plans to provide advanced algorithms and new header formats, along with support for automatic key exchange and refresh, also on client platforms later this year.

The following products are covered by the table below:

- AIX IPSec Client (supplied with the eNetwork Firewall for AIX)
- Windows 95 IPSec Client (supplied with the eNetwork Firewall for AIX)

- OS/2 TCP/IP V4.1 IPSec Client (part of the OS/2 TCP/IP V4.1 stack)
- Windows 95 eNetwork Communications Suite V1.1

Table 2. Client Platforms - Supported VPN Features

Feature		AIX IPSec Client	Windows 95 IPSec Client	OS/2 TCP/IP V4.1	Windows 95 Comms Suite
Tunnel Type	Manual	√		√	√
	Dynamic		√	√	
IPSec Protocol	AH	√	√	√	√
	ESP	√	√	√	√
Header Format	RFCs 18xx	√	√	√	√
	New Draft Headers				
SA Type	Transport Mode			√	√
	Tunnel Mode	√	√	√	√ (ESP only)
AH Transform	Keyed MD5	√	√	√	√
	HMAC MD5				√
ESP Transform	DES CBC 4	√	√		√
	DES CBC 8	√	√		√
	CDMF	√	√	√	
Connectivity	LAN	√		√	√
	Dial-up		√	√	√
Other	Packet Filters	limited		√	
	Logging			√	
	IP Version 6				√

4.3.1 AIX IPSec Client (Supplied with the eNetwork Firewall for AIX)

Because the functionality found on an AIX V4.3 system is much better than the one provided in the AIX IPSec Client we recommend you use AIX V4.3 whenever possible. This means use the AIX IPSec Client only if you have to use an AIX level below V4.3. Therefore we did not assign a high priority to testing this client. Please see 4.4, “Interoperability between the IBM Solutions” on page 75 for references.

4.3.1.1 VPN Features

The AIX IPSec Client offers:

- Tunnel types: Manual
- SA Types: Tunnel
- IPSec protocols: All (AH and ESP)
- Header Formats: RFCs 18xx
- Policies: All (auth, encr, encr/auth, auth/encr)
- AH transforms: Keyed MD5
- ESP transforms: DES CBC 4, DES CBC 8 and CDMF

- Connectivity: LAN
- Other VPN-related features: Limited packet filters

Please refer to section 7.4 of redbook *Protect and Survive Using IBM Firewall 3.1 for AIX, SG24-2577* for a description of how to install the AIX IPSec Client.

4.3.2 Windows 95 IPSec Client (Supplied with the eNetwork Firewall for AIX)

This client is exclusively used with the eNetwork Firewall for AIX. It has two big advantages:

- No requirement for a static IP address on the client.
- eNetwork Firewall for AIX provides it free of charge.

If the above is not important to you or if you need LAN connectivity, the eNetwork Communications Suite might be a better alternative.

4.3.2.1 VPN Features

The Windows 95 IPSec Client includes the following VPN features:

- Tunnel types: Dynamic
- SA Types: Tunnel
- IPSec protocols: All (AH and ESP)
- Header Formats: RFCs 18xx
- Policies: All (auth, encr, encr/auth, auth/encr)
- AH transforms: Keyed MD5
- ESP transforms: DES CBC 4, DES CBC 8 and CDMF
- Connectivity: Dial-up
- Other VPN-related features: None

4.3.2.2 Remote Dial-Up

The client does not tie you to a specific PPP server. The IP address that is assigned by your ISP is irrelevant. The client can support any dial-up Internet provider that offers support for the Password Authentication Protocol (PAP) or the Microsoft Challenge Handshake Authentication Protocol (MS-CHAP). CompuServe is not supported by the Windows 95 IPSec client.

You can change PPP server and IP addresses and it does not affect the operation of the Windows 95 IPSec Client. Other vendors are sensitive to the specific TCP/IP address; if you change the address, you must reconfigure your client.

In order to establish a connection you first dial into your Internet Service Provider (ISP). Then you log on to the firewall. An encrypted secure login is provided for the Windows 95 IPSec Client using Secure Sockets Layer (SSL) Version 2 technology.

After the tunnel is established, all IP traffic flows through the tunnel. Once users make a connection, they have full TCP/IP access to whatever servers are behind the firewall and can use, among others, FTP, telnet, HTTP, and mail applications.

When using the Windows 95 IPSec client, set the domain name server entry to the server responsible for the name resolution of the the secure network.

Otherwise you cannot resolve host names of that network (unless official IP addresses are used within the secure network).

4.3.2.3 eNetwork Firewall for AIX V3.2

The Windows 95 IPSec client in firewall V3.2 has changed as follows:

1. Strong user authentication is now available (it now supports the same security mechanism as the firewall) using Security Dynamics ACE/SecurID cards and tokens.
2. Redesigned client interface.
3. More status information.

We used the Windows 95 IPSec Client as well as the OS/2 TCP/IP V4.1 IPSec Client for our remote dial-in scenario, but we described only the setup of the OS/2 client (see 7.3, “Configuring the Components” on page 137). Please refer to section 7.7 of redbook *Protect and Survive Using IBM Firewall 3.1 for AIX, SG24-2577* for a description of how to install the Windows 95 IPSec Client.

4.3.3 OS/2 TCP/IP V4.1 IPSec Client

The VPN implementation is very similar to the Windows 95 IPSec Client as far as the dynamic tunnel type is concerned. Using this feature, the OS/2 TCP/IP V4.1 IPSec Client acts as a tunnel partners to the eNetwork Firewall for AIX. The client is not provided with the firewall but with the OS/2 TCP/IP V4.1 stack.

There is no interface provided to configure manual tunnels with the OS/2 TCP/IP V4.1 IPSec Client. However, you can manually change the tunnel definition, policy and filter files to facilitate that. In fact, the OS/2 IPSec kernel could then behave like a mini-firewall supporting manual tunnels. For details on how to enable that additional functionality, please see 9.4, “Additional OS/2 TCP/IP V4.1 IPSec Client Combinations” on page 179.

TCP/IP V4.1 for OS/2 is available as an update via Software Choice from the following Web Site:

```
http://service.software.ibm.com/asd-bin/doc/index.htm
```

Note: TCP/IP V4.1 for OS/2 is supported on OS/2 Warp 4, OS/2 Warp Server and OS/2 Warp Server SMP. It requires the following components to be installed prior to its own installation:

1. OS/2 Feature Installer V1.2.1, or higher
2. Netscape Navigator V2.02e for OS/2, or later
3. Java V1.1.4 for OS/2

All of those components are also available via Software Choice.

4.3.3.1 VPN Features

The OS/2 TCP/IP V4.1 IPSec Client offers:

- Tunnel types: Dynamic, manual
- SA Types: Tunnel, transport
- IPSec protocols: All (AH and ESP)
- Header Formats: RFCs 18xx
- Policies: All (auth, encr, encr/auth, auth/encr)
- AH transforms: Keyed MD5

- ESP transforms: CDMF

  Note: The OS/2 TCP/IP V4.1 IPSec Client has been designed to also support DES CBC 4 and DES CBC 8 transforms, but because TCP/IP V4.1 is available as an update via the World Wide Web for which there is no reliable tracking of the origin of a request, the strong encryption module has been removed to comply with U.S. government export restrictions.

- Connectivity: LAN and Dial-up
- Other VPN-related features: Packet filtering and logging

4.3.3.2 Packet Filters

Packet filtering is required to drive the data through the IPSec device drivers. Full packet filtering is possible but only a subset of this function is exposed by TCP/IP V4.1 via the VPN configuration panels. See 9.4.1, “Configuring IPSec Filters and Tunnels” on page 180 for more information on packet filtering.

4.3.3.3 Domain Name Resolution

The client offers a nice feature to solve the problem of how to resolve names in the remote secure network. There are extra fields on the configuration panel to specify the domain and server name for the remote secure network.

You can find an example for an OS/2 TCP/IP V4.1 IPSec Client configuration in 7.3, “Configuring the Components” on page 137.

4.3.4 Windows 95 eNetwork Communications Suite V1.1

This client offers an implementation of the manual tunnel type for both a LAN and a dial-up environment. These are also the the two main differentiators to the Windows 95 IPSec Client supplied by the eNetwork Firewall for AIX, which implements the dynamic tunnel in a dial-up environment only.

4.3.4.1 Versions

There are two versions of the product:

1. International export version
2. U.S. export controlled version

Because DES is the only supported ESP transform with this client, there is no encryption in the international export version since DES is generally not allowed outside the U.S. and Canada.

4.3.4.2 VPN Features

The eNetwork Communications Suite offers:

- Tunnel types: Manual
- SA Types: All (tunnel and transport mode)
- IPSec protocols: All (AH and ESP)
- Header Formats: RFCs 18xx
- Policies: All important ones (auth, encr, encr/auth)
- AH transforms: Keyed MD5 and HMAC MD5
- ESP transforms: DES CBC 4 and DES CBC 8
- Connectivity: LAN and Dial-in

- Other VPN related features: IPv6

4.3.4.3 Hints

All values entered on the configuration panel are hexadecimal numbers. This includes SPIs which are for instance decimal on an AIX V4.3 system. Therefore you will have to convert the SPIs to decimal numbers before entering them for instance on the AIX V4.3 system.

You will have to reboot the Windows 95 system whenever a new tunnel is added or an existing tunnel is changed.

You can find an example for an eNetwork Communications Suite configuration in 6.4.2, "Configuration of the eNetwork Communications Suite Client" on page 127.

4.4 Interoperability between the IBM Solutions

Summing up the details about the different IBM eNetwork VPN solutions, Figure 34 illustrates a compound scenario of how those solutions could be best deployed in a real-world virtual private network.

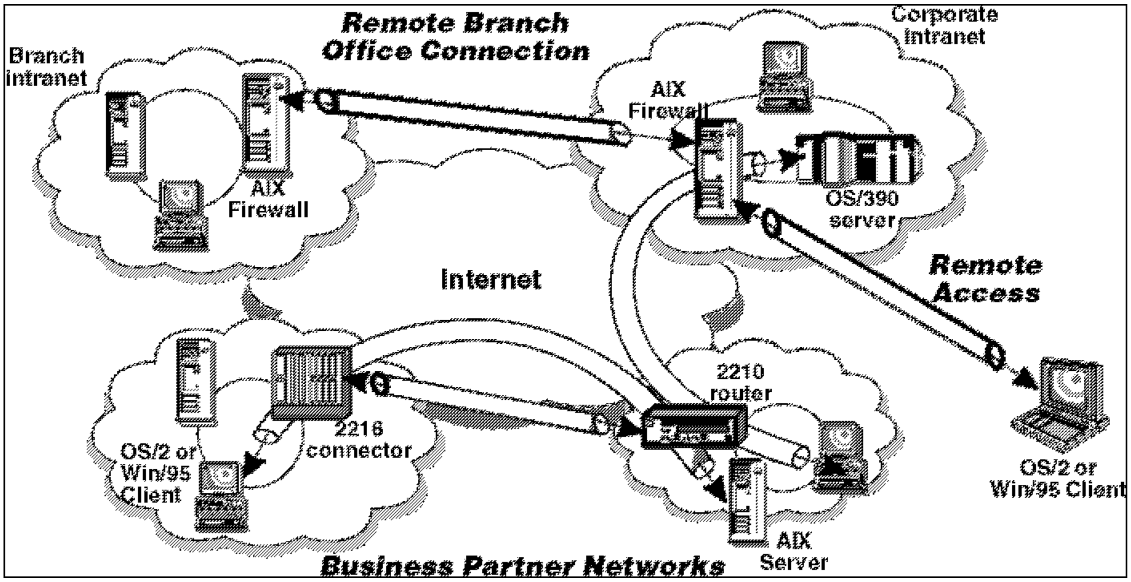


Figure 34. IBM eNetwork VPN Interoperability

Because there are many different possibilities for how to establish VPNs between any two of the mentioned IBM products the next table shows our general recommendations on how to configure a connection between two specific IBM eNetwork VPN products.

Note: In cases where more than one configuration is possible, for instance between the eNetwork Firewall for AIX and AIX V4.3 (I,M), we have only listed the configuration that we would prefer.

The characters used in Table 3 on page 76 have the following meanings:

- I: IBM Tunnel
- M: Manual Tunnel
- D: Dynamic Tunnel

The absence of a character in a field means that we were not able to find a configuration that worked between the two products.

Table 3. IBM Products - Interoperability Recommendations

	eNetwork Firewall for AIX	**AIX V4.3**	**2210 / 2216 / 3746**	**OS/390 Server**	**AIX IPSec Client**	**Win95 IPSec Client**	**eNetwork Comms Suite**	**OS/2 TCP/IP V4.1**
eNetwork Firewall for AIX	I	I		M	M	D	M	D
AIX V4.3	I	I	M	M	M		M	M
2210 / 2216 / 3746		M	M					
OS/390 Server	M	M		M	M		M	M
AIX IPSec Client	M	M		M	M		M	M
Windows 95 IPSec Client	D							
eNetwork Communications Suite	M	M		M	M		M	M
OS/2 TCP/IP V4.1 IPSec Client	D	M		M	M		M	M

4.5 Recommended Configuration Reference

The following sections contain references to more detailed information on how to configure a setup between two IBM products of your choice.

4.5.1 eNetwork Firewall for AIX

See the following references depending on what your partner is:

- eNetwork Firewall for AIX: Chapter 5, “Branch Office Connection Scenario” on page 81.
- AIX V4.3: 9.1.2, “Interoperability between eNetwork Firewall for AIX and AIX V4.3” on page 173.
- 2210/2216 Router: Currently not possible - see 4.5.5, “Other Combinations and Future Implementations” on page 78.
- OS/390 Server: 9.3.1, “Interoperability with the eNetwork Firewall for AIX” on page 174.
- AIX IPSec Client: Section 7.4 of redbook *Protect and Survive Using IBM Firewall 3.1 for AIX*, SG24-2577.
- Windows 95 IPSec Client: Section 7.7 of redbook *Protect and Survive Using IBM Firewall 3.1 for AIX*, SG24-2577.
- eNetwork Communications Suite: 9.5.1, “Interoperability with the eNetwork Firewall for AIX” on page 189.
- OS/2 TCP/IP V4.1 IPSec Client: 7.3, “Configuring the Components” on page 137.

4.5.2 AIX V4.3

See the following references depending on what your partner is:

- eNetwork Firewall for AIX: 9.1.2, "Interoperability between eNetwork Firewall for AIX and AIX V4.3" on page 173.
- AIX V4.3: 9.2.1, "Interoperability between Two AIX V4.3 Systems" on page 174.
- 2210/2216 Router: Was not tested.
- OS/390 Server: 9.3.2, "Interoperability with AIX V4.3" on page 176.
- AIX IPSec Client: Was not tested.
- Windows 95 IPSec Client: Currently not possible - see 4.5.5, "Other Combinations and Future Implementations" on page 78.
- eNetwork Communications Suite: 6.4, "Configuration of the Nested Tunnel between the Tunnel Endpoints" on page 120.
- OS/2 TCP/IP V4.1 IPSec Client: 9.4.4, "Interoperability between OS/2 TCP/IP V4.1 IPSec Client and AIX V4.3" on page 187.

4.5.3 2210/2216 Router

See the following references depending on what your partner is:

- eNetwork Firewall for AIX: Currently not possible - see 4.5.5, "Other Combinations and Future Implementations" on page 78.
- AIX V4.3: Was not tested.
- 2210/2216 Router: Redbook *A Comprehensive Guide to Virtual Private Networks, Volume II: IBM Nways Router Solutions*, SG24-5234, to be published later this year.
- OS/390 Server: Currently not possible - see 4.5.5, "Other Combinations and Future Implementations" on page 78.
- AIX IPSec Client: Currently not possible - see 4.5.5, "Other Combinations and Future Implementations" on page 78.
- Windows 95 IPSec Client: Currently not possible - see 4.5.5, "Other Combinations and Future Implementations" on page 78.
- eNetwork Communications Suite: Currently not possible - see 4.5.5, "Other Combinations and Future Implementations" on page 78.
- OS/2 TCP/IP V4.1 IPSec Client: Currently not possible - see 4.5.5, "Other Combinations and Future Implementations" on page 78.

4.5.4 OS/390 Server

See the following references depending on what your partner is:

- eNetwork Firewall for AIX: 9.3.1, "Interoperability with the eNetwork Firewall for AIX" on page 174.
- AIX V4.3: 9.3.2, "Interoperability with AIX V4.3" on page 176.
- 2210/2216 Router: Currently not possible - see 4.5.5, "Other Combinations and Future Implementations" on page 78.
- OS/390 Server: Redbook *Stay Cool on OS/390: Installing Firewall Technology*, SG24-2046.

- AIX IPSec Client: Section 5.4 of redbook *Stay Cool on OS/390: Installing Firewall Technology*, SG24-2046.
- Windows 95 IPSec Client: Currently not possible - see 4.5.5, “Other Combinations and Future Implementations.”
- eNetwork Communications Suite: 9.3.3, “Interoperability with the eNetwork Communications Suite” on page 176.
- OS/2 TCP/IP V4.1 IPSec Client: 9.4.3, “Interoperability between OS/2 TCP/IP V4.1 IPSec Client and OS/390 Server” on page 186.

4.5.5 Other Combinations and Future Implementations

IBM has announced that it will provide IPSec capabilities on the Windows NT and AS/400 platforms later this year. IBM has also announced that it will provide automatic key exchange and key refresh functionality, based on ISAKMP/Oakley, on all its eNetwork VPN products; first implementations are to be expected later this year. At that time, there will be standard-compliant IPSec connection possibilities between all of IBMs eNetwork VPN solutions.

Future enhancements will also be described in a separate redbook, *A Comprehensive Guide to Virtual Private Networks, Volume III: Secure Key Exchange and Policy Management*, early next year.

4.6 Interoperability between IBM VPN Solutions and Other Vendors

If you talk about VPNs, particularly if your VPN extends to your business partners, you may be confronted with the issue of interoperability: does your VPN equipment work with other vendors′ VPN-enabled products? While the IPSec standards go a long way towards fostering interoperability, the fact that two different implementations conform to IPSec doesn′t necessarily mean that they will interoperate with one another. No standard ever nails down all the details, and like most standards, IPSec provides for a mix of mandatory and optional features.

The IPSec Bakeoffs provide a means for vendors to informally test their IPSec implementations′ interoperability, and to iron out any differences in interpretation of the relevant IPSec standards. The most recent IPSec Bakeoff was held in Raleigh, NC, in March 1998. IBM participated with its eNetwork Firewall for AIX, AIX V4.3, OS/390 Server and 2210/2216 Router products.

IBM VPN solutions interoperated successfully with implementations from many vendors, such as: Frontier Technologies, CIsco, Ascend, Watchguard, Cylan, NIST, Red Creek, Ashley Laurent, Toshiba, Hewlett-Packard, Radguard, Sun, 3Com, Intel, V-One, SEI, Secure Computing, Internet Devices, and Interlink.

Bakeoffs are informal affairs, and the functions provided for testing may be at any stage of development, from early prototypes, to beta code, to fully released commercial products. Testing is usually done in pairs, and each vendor decides what functions it considers most important to evaluate. The results of the informal Bakeoff testing are then fed back to the IPSec Working Group, which uses that input as the basis for refining and clarifying the IPSec protocol standards.

Part 3. VPN Scenarios and Implementation

Chapter 5. Branch Office Connection Scenario

This chapter shows the most common use of VPN technology, the secure connection of two trusted intranets over the public Internet. The focus in this scenario is on protecting your intranets from outside attacks and securing corporate data flowing on the Internet.

This scenario can be also deployed in one intranet. It may be reasonable to connect in this way, for example, two highly secure development laboratory networks over the existing corporate network infrastructure.

Consider a company that was running its own private network, using its own routers, bridges, and private lines. If the company had campuses at geographically dispersed sites, it may prove more economical to break the corporate network into pieces (the intranets), add a firewall to control traffic flow across the intranet/Internet boundary, and then procure service from one or more ISPs to interconnect the intranets over the Internet backbone.

For this example, we assume that company A just wants to enable communication between its intranets, but does not necessarily want to communicate with entities outside of company A.

5.1 Design Considerations

Let us consider how company A could construct a virtual private network for interconnecting its intranets securely. In the discussion we do not take into account the basic Internet access issues, since these can be well separated and are outside the scope of this redbook.

5.1.1 Authenticating Backbone Traffic

The Internet will be carrying traffic not just from company A′s VPN, but also from other VPNs. Company A′s firewalls must admit only traffic from company A′s VPNs and must reject all other incoming traffic. They might admit non-VPN incoming traffic destined to them in case they provide general Internet connectivity, for example, if they operate proxies or SOCKS servers. However, in the case of a large company with many VPNs it is worth considering the separation of functions, that is, dedicated security gateways for VPNs and others for general Internet access. It is more expensive but dedicated VPN gateways are much harder to bring down by denial of service attacks, because they accept only authenticated traffic. Companies in most cases are much more sensible to the loss of branch connectivity than to the loss of Web access.

Deploying IPSec′s authentication protocols in company A′s firewalls (or IPSec-enabled routers) at the intranet boundary will accomplish these goals. IPSec′s authentication techniques are cryptographically strong, so they provide significantly better protection against address spoofing and denial of service attacks than would rely on conventional, non-cryptographic filtering techniques. In this scenario, cryptographic authentication using HMAC will be the first line of defense. Having established that the traffic has come from somewhere within company A′s network, non-cryptographic filtering can then be used as the second line of defense to provide more granular access control, if desired.

5.1.2 Data Confidentiality

It should be obvious that company A will want to keep its data confidential (that is, encrypted) while it is in transit across the public Internet. But it is not always clear if the data should also be protected when it flows within its own intranets. If the company had not previously encrypted its internal traffic when it used a monolithic private network, it may not see value in encrypting it when it flows within its intranets.

If a company does not believe that it is subject to internally mounted attacks, the simplest solution will be to encrypt and authenticate traffic flowing between firewalls, and make no security-related changes to the end systems themselves. This has the advantage of much fewer security associations to manage: two per firewall for bidirectional data flow, compared to two per host for host-to-host encryption. But it has the disadvantage that traffic is exposed to relatively simple attacks while it flows in the intranet. Since authentication is also needed between firewalls, the simplest branch office VPN will use ESP in tunnel mode with authentication between the two firewalls. Another solution is the combination of AH and ESP in tunnel mode, which has the advantage of authentication of the outer IP header as well, thus avoiding the denial of service attacks. The latter is the only possibility to provide both authentication and encryption when the firewall product does not yet support the latest IPSec specifications, such as the current releases of the IBM eNetwork Firewall.

This is the situation described in the 3.4.2, “Case 2: Basic VPN Support” on page 52 in Chapter 3, “Description of IPSec” on page 39.

For considerations on how to configure a VPN solution between branch offices that can protect you against threats both in the Internet and in your company′s intranet as well, please refer to Chapter 6, “Business Partner/Supplier Network Scenario” on page 111 where we discuss this topic in the slightly different context of two different companies. However, the solution is the same.

5.1.3 Addressing Issues

We assumed that company A previously had a traditional network in place, where its various intranets were interconnected over private facilities, such as leased lines or frame relay. We also assumed that company A has already developed an address plan for its network. Since the network was self-contained and the backbone used only private facilities, company A could have used either globally ambiguous (private) IP addresses (that is, of the form 10.x.y.z) or globally unique (public) addresses obtained from the Network Information Center (NIC).

Because assignment of public IP addresses is coordinated through a global authority, they are unambiguous. Public addresses are routable everywhere. However, because private address assignments are facilitated locally without coordination by a global authority, they are ambiguous when used in the public Internet; they are routable only within a company′s own private network.

In summary:

1. If company A uses public addresses in its network, the addresses can continue to be used without change in the VPN environment. If it is desired to hide them while the datagram is in transit over the Internet, an ESP tunnel can be used between firewalls.

2. If company A uses private IP addresses in its network, the addresses can also continue to be used on all subnets that have no physical connection to the public Internet. But for those subnets that do connect to the public Internet, typically the exit links at the boundary of the intranet, a public IP address must be used.

ESP tunnel mode or AH and ESP in tunnel mode between firewalls handles both situations. The tunnel′s new IP header will use the global addresses of the two firewalls, allowing datagrams to be routed over the Internet between the two firewalls (or routers). The header of the original (inner) IP datagram will use the IP addresses assigned for use in the intranet; since these addresses will be hidden from view by ESP′s encryption protocol, they can be either publicly or privately assigned.

5.1.4 Routing Issues

Because a VPN in fact resembles a set of IP networks, all but the smallest VPNs will typically need to deploy an IP routing protocol between the gateway machines (firewalls or routers) at the boundaries of the company′s intranets. Routing protocols typically exchange information that will describe the topology of the VPN. That is, the topology updates will describe the IP addresses that are reachable within each intranet that participates in the VPN. IPSec can be used to both encrypt and authenticate the routing information, thus hiding topological details of the intranet as they are exchanged across the public network.

The company′s network administrator(s) can incorporate conventional IP routing protocols into the firewalls, and then use IPSec protocols to encrypt and authenticate the exchange of routing information among the firewalls. Figure 35 illustrates this concept schematically for a sample configuration that consists of three branch offices of a given company that need to communicate among themselves via the public Internet.

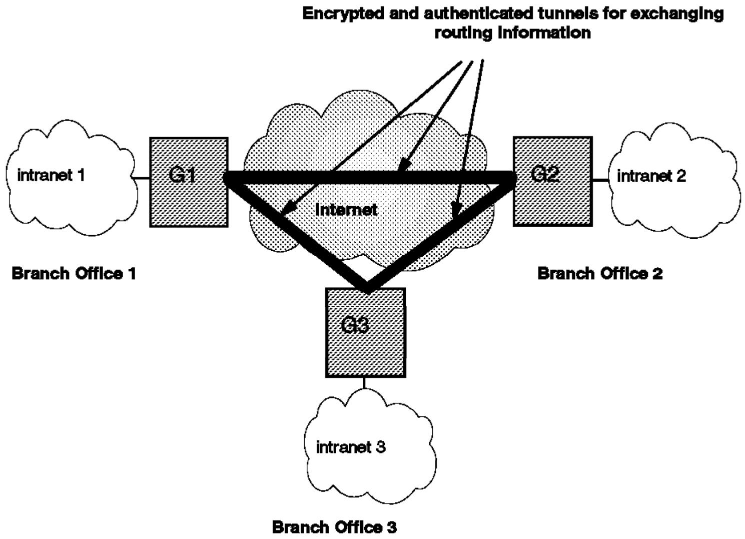


Figure 35. Exchanging Routing Information Securely

When an IPSec tunnel is established between a pair of firewalls, they appear to be logically adjacent to one another, even though there may be several routers along the actual physical path. Each pair of virtually adjacent security gateways will set up a security association between themselves, using ESP in tunnel mode with authentication or AH and ESP in tunnel mode to provide both encryption and authentication. The routing information that is exchanged will then be hidden from view because it will have been encrypted.

Because the set of firewalls participate in a common routing protocol, they will know the correct firewall for reaching any given destination host within the intranets. Hence, traffic arriving at an exit firewall can be sent via an ESP tunnel, using its authentication option or a combined AH-ESP tunnel, and can then be authenticated by the entry firewall that protects the remote branch office′s Intranet.

Thus, IPSec makes it possible to exchange routing information and user data between branch offices over the Internet while preserving the confidentiality of both user data and intranet topology information. Because routing information (for example, IP addresses) is visible only to other members of the corporate network, this scheme can be used regardless of whether addresses used in the interior of an intranet are globally unique or privately assigned.

To be more specific, since the intranet addresses are carried within encrypted routing update messages, they are neither visible to, nor used by, any of the routers in the Internet. Therefore, if company A′s intranets use a self-consistent addressing scheme, either public or private, network address translation is not needed for intranet addresses. Encapsulating encryption already hides the interior addresses, and all backbone routing is based only on the public IP addresses of the boundary gateways. Finally, depending on the sophistication of the routing algorithms, it may also be possible to support redundant entry/exit points into a corporate network.

5.1.5 Summary: Branch Office Connection

This application replaces existing private lines or leased lines in a corporate network, and uses the public Internet as the backbone for interconnecting a company′s branch offices. (This solution is not limited only to branch offices, but can also be be applied between any collection of a company′s geographically dispersed sites, such as labs, manufacturing plants, warehouses, etc.) This solution does not mandate any changes in the clients (PCs or servers) unless it is desired to protect against internal attacks as well as external ones.

The design features are:

- Client machines (hosts and servers) need not support IPSec if the intranets are considered to be trusted and secure. This minimizes the migration issues of moving to a VPN approach and maintains the pre-existing host-to-host security policies and procedures of the original network. IPSec support will be required only at the intranet boundaries, that is, in the VPN gateway boxes. Also there will be no security-related protocols required in any of the routers, bridges, or switches that are located either in the interior of the intranets or in the public backbone Internet.

- VPN gateways situated at the perimeter of each branch office intranet implement the basic firewall functions (for example, packet filtering) and also support IPSec protocols to build secure and authenticated tunnels between all VPN gateways of the branch office networks that comprise the VPN.

  User data traffic will be both authenticated and encrypted. Any inbound traffic that can not be authenticated by the VPN gateway will not be delivered into the intranet. Authentication will be cryptographically based, using AH or the authentication option for IPSec′s ESP protocol.

- Routing control messages will be exchanged among the set of VPN gateways, and these messages will also be encrypted and authenticated using IPSec procedures.
- If the number of VPN gateways in the initial VPN deployment is small, key distribution and security association definition can be handled by manual methods. But as the VPN′s size grows to encompass more and more intranets, the automated IBM tunnel or ISAKMP/Oakley procedures will rapidly become a necessity.
- Security associations will be set up among the set of VPN gateways. Because the source and destination hosts (that is, clients and servers) are not required to support IPSec, no security associations need to be set up between hosts, and no keys need to be assigned to them. In the future, if even stronger security is desired for host-to-host communications, then clients and servers will need to support the IPSec protocols.
- The IP addresses assigned for use in the intranets can be used as is, regardless of whether they were assigned from a public or a private address space. Only the interfaces of the VPN gateways that attach to the Internet backbone are required to use globally unique IP addresses.
- Packet filtering rules, if any, that were used in the pre-VPN network should be installed on the VPN gateway to control traffic that enters the branch office intranet. They can be used as a second line of defense, after the packets have been authenticated by IPSec′s AH protocol.
- If end-to-end IPSec functions are deployed between hosts, then new packet filtering rules will be needed in the firewalls to recognize the IPSec AH and ESP headers.

5.2 Scenario Setup

We built the test network shown in Figure 36 on page 86. The Internet is simulated by the ITSO internal itso.ral.ibm.com subnet, to which we attached our private subnets representing the company′s different sites.

Note: The same infrastructure is used throughout the scenarios, which is why the naming conventions do not reflect precisely the actual situation.

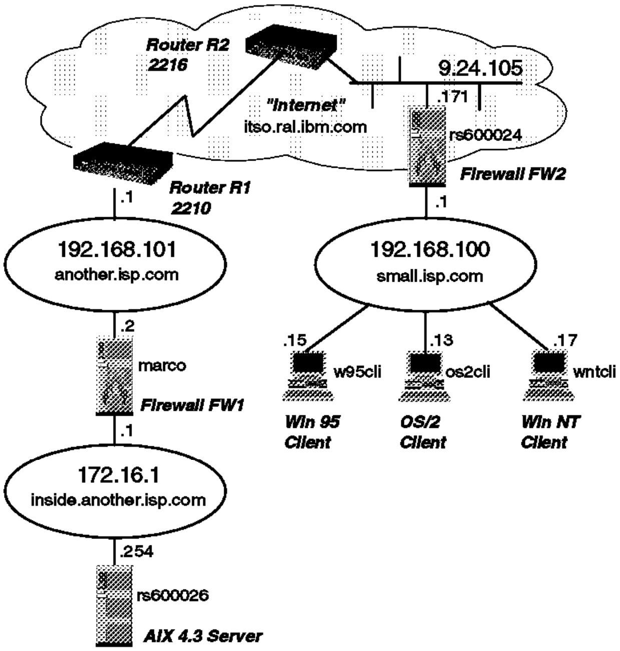


Figure 36. Branch Office Connection Test Network

The subnets another.isp.com and inside.another.isp.com correspond to the company′s main site, which is attached to the Internet via the router R1. The outer subnet another.isp.com is a *demilitarized zone (DMZ)* or border network and it is used for added security and for a place from where to provide publicly accessible services, typically the company′s Web site and anonymous FTP site. The servers providing those services are not shown, since they are irrelevant to our VPN scenario.

The firewall FW1 (marco) is the only gateway to the protected internal subnet inside.another.isp.com. In this subnet reside the company′s servers that run business-critical applications. These must be made accessible to the branches. In our test lab, the application server is the AIX 4.3 machine rs600026. We access it from the branch via telnet sessions. The server is IPSec-enabled, but there is no need to use its IPSec capabilities in this scenario.

Both firewall machines run AIX V4.2.1 and eNetwork Firewall for AIX V3.1.1.5. We have also tested the same scenario successfully using AIX V4.3.0 and eNetwork Firewall for AIX V3.2.1 on both machines.

Note: We do not elaborate on the general setup of the firewall that provides protection against intruders from the Internet, since this topic is thoroughly covered in the *Protect and Survive Using IBM Firewall 3.1 for AIX*, SG24-2577 redbook.

The branch office subnet small.isp.com connects directly to the internet via the firewall FW2 (rs600024). In real life there is usually a router as well, but from the VPN point of view it is transparent. The clients represent the current choice of most common client operating systems: Windows 95, Windows NT, and OS/2. They do not run IPSec protocols, so no modification is required for them.

As far as the tunnels are concerned, we build an authenticated and encrypted tunnel between the firewalls marco and rs60024 and direct all traffic between the private subnets into this tunnel. This is outlined in Figure 37 on page 87.

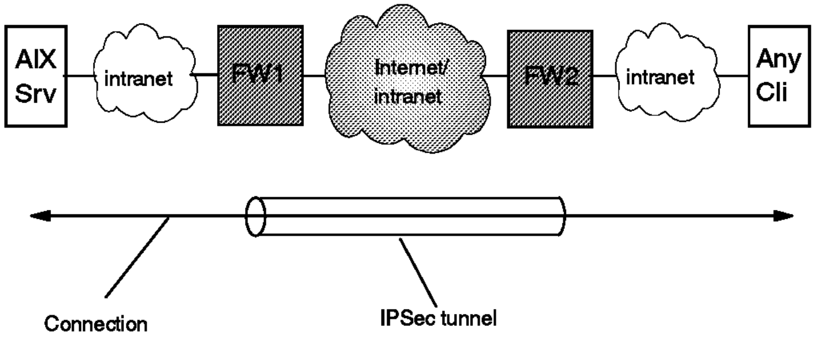


Figure 37. Branch Office Connection Scenario - Tunnel Between Firewalls

Note: As a cost-effective alternative, FW2 could be replaced by a small IPSec-enabled router in a customer environment.

Please read the following sections to learn more about the details.

5.3 Configuration of the Firewalls

The following sections guide you through the configuration of the branch office scenario. After discussing necessary prerequisites we provide a summary of the configuration steps followed by the detailed panel by panel procedure. Further to the end of the chapter we have included information to help you to understand the background behind the tunnel related filter rules (″how does it work?″) and we close by discussing variations of the branch office scenario. We assume that you are familiar with the information presented in Chapter 4, “IBM eNetwork VPN Solutions” on page 59.

5.3.1 Prerequisite Steps

The following prerequisites need to be met at both firewall sides.

1. Basic firewall installation and setup has been completed (including network objects for your non-secure firewall interface and for your secure network).
2. Update your firewall code to the latest level. Our tests were conducted with V3.1.1.5 and also with V3.2.1, and we had no problems. Be aware that there are known VPN problems with older levels of code.
3. Ensure that IP forwarding is enabled on both firewalls (using the followingcommand: `no -o ipforwarding=1`). Otherwise the packets cannot be routed between the secure and non-secure interfaces of the firewall.

   Good to know

   When using eNetwork Firewall for AIX V3.2.1, IP forwarding is enabled by default.

4. You need coherent IP addresses in both secure networks. For example, you cannot use the same private IP addresses on both intranets. See 5.1.3, “Addressing Issues” on page 82 for details.
5. Make sure that the IP routing tables in the clients are correct. They need to point to the firewall for addresses in the remote secure network. In general routing has been discussed in 5.1.4, “Routing Issues” on page 83.
6. You will probably need to resolve domain names of hosts located in the remote secure network. In this case your DNS server must be able to

resolve DNS requests for hosts belonging to the remote domain or your clients must be able to resolve the remote addresses.

Note: If you need more information on how to set up the eNetwork Firewall for AIX we recommend the redbook *Protect and Survive Using IBM Firewall 3.1 for AIX*, SG24-2577.

5.3.2 Summary of the Necessary Steps

This section provides a high-level summary of the necessary steps to set up this scenario. It is provided for the convenience of users who are experienced in using the AIX firewall configuration panels and familiar with the general concepts of setting up network objects, connections and services. Users who are new to these tasks, please read a detailed description in 5.3.3, “Firewall Setup Details” on page 90.

In order to get a tunnel to work you need to accomplish three task groups on each firewall:

1. Create the necessary network objects
2. Define the tunnel
3. Define the packet filters necessary to allow traffic through the tunnel

Note: The order of the tasks is important. The objects contain the IP addresses needed for the tunnel definition, which in turn includes a tunnel ID. The tunnel ID is needed as a parameter for the packet filters so that the filters know that the packets are to be sent to the IPSec kernel, which in turn processes them according to the definition of the tunnel specified in the tunnel ID.

For the following summary we named the firewalls FW1 and FW2. The steps for the basic tunnel setup are:

5.3.2.1 Firewall FW1 (Tunnel Owner)

1. Create two new network objects:
   - Non-secure interface of remote firewall FW2 (type: firewall)
   - Secure network of remote firewall FW2 (type: network)
2. Create the tunnel definition. Choose tunnel type **IBM tunnel**. For a description of the other fields see 4.1.2.2, “IBM Tunnel” on page 62.
3. Export the tunnel definition. The directory you specify must already exist.
4. Add the connection for tunnel traffic between the two firewalls:
   - Source object: Non-secure interface of firewall FW1
   - Destination object: Non-secure interface of (remote) firewall FW2

   Services to be included:

   - VPN encapsulation
   - VPN key exchange
5. Now you have to create a new service needed for the second connection that covers the traffic between the secure networks. Click on **Services** and copy the predefined service VPN traffic 2/2 to a new service called, for instance, VPN traffic 2/2 tunnel xx. The only field you have to change is Override Tunnel ID. Set it to xx by using the **Select** button (xx being the tunnel ID you have specified in the tunnel definition).

6. Add the connection for the traffic between the endpoints of the tunnel:

   - Source object: Secure network of firewall FW1
   - Destination object: Secure network of firewall FW2

   Services to be included:

   - VPN traffic 1/2
   - VPN traffic 2/2 tunnel xx

7. Activate the new packet filter ruleset.

The tunnel setup of firewall FW1 is now finished. Transport the exported files from your firewall (FW1) to the remote firewall (FW2). See 6.6, “Manual Key Distribution” on page 131 for some suggestions.

5.3.2.2 Firewall FW2 (Tunnel Partner)

1. Create two new network objects:

   - Non-secure interface of remote firewall FW1 (type: firewall)
   - Secure network of remote firewall FW1 (type: network)

2. Import the tunnel definition from firewall FW1. The import will automatically switch local and remote addresses (as local and remote are relative to the specific firewall).

   Note: The import will also work if you did not create the objects in step 1 and 2. But you need the objects for the connections in the next steps.

3. Add the connection for tunnel traffic between the two firewalls:

   - Source object: Non-secure interface of firewall FW2
   - Destination object: Remote firewall FW1

   Services to be included:

   - VPN encapsulation
   - VPN key exchange

4. Now you have to create a new service needed for the second connection that covers the traffic between the secure networks. Click on **Services** and copy the predefined service VPN traffic 2/2 to a new service called, for instance, VPN traffic 2/2 tunnel xx. The only field you have to change is Override Tunnel ID. Set it to xx by using the **Select** button (xx being the tunnel ID you have specified in the tunnel definition).

5. Add the connection for the traffic between the endpoints of the tunnel:

   - Source object: Secure network of firewall FW2
   - Destination object: Secure network of firewall FW1

   Services to be included:

   - VPN traffic 1/2
   - VPN traffic 2/2 tunnel xx

6. Activate the new packet filter ruleset.

A last step is now required on both firewalls, the activation of the tunnel. Wait some time (30 seconds should do) for the initialization process to finish. Now your tunnel is ready for traffic. See 5.4, “Testing the Tunnel” on page 103 on how to test the tunnel.

5.3.3 Firewall Setup Details

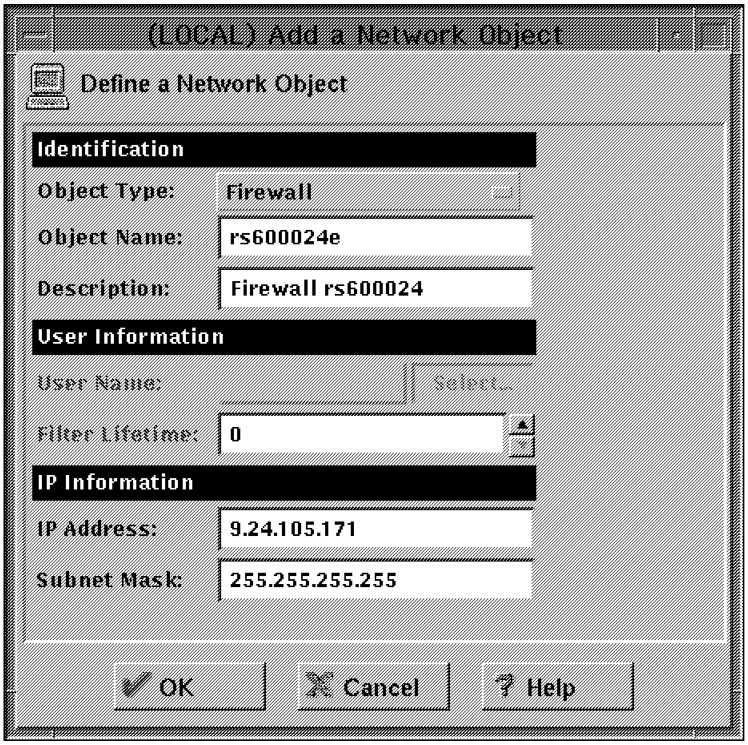
This section explains the above summary in more detail, but the steps remain the same. The name of firewall FW1 in our environment was marco; firewall FW2 was named rs600024. The details about our environment have been described in 5.2, "Scenario Setup" on page 85.

5.3.3.1 Prerequisites

Before starting the tunnel configuration please make sure that you have met the prerequisites discussed in 5.3.1, "Prerequisite Steps" on page 87 on both firewalls.

5.3.3.2 Creating the Network Objects in the First Firewall

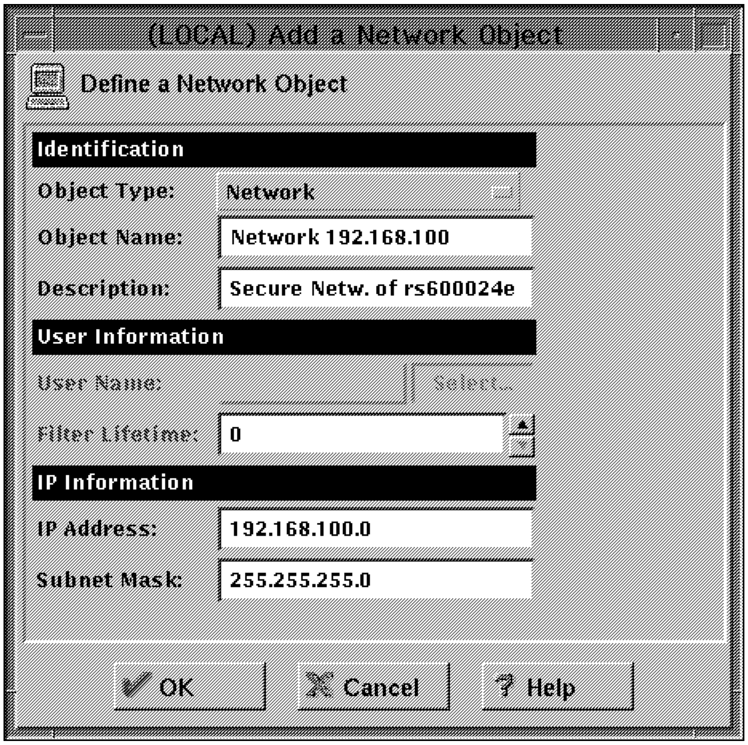
We started the setup on the first firewall (marco) by creating the network object for the non-secure interface of the remote firewall (rs600024e). This object serves as destination address in the tunnel definition and in the first connection that we create in a later step. Take care that the object type is firewall.



Object Type: Firewall
Object Name: rs600024e
Description: Firewall rs600024
User Name:
Filter Lifetime: 0
IP Address: 9.24.105.171
Subnet Mask: 255.255.255.255

Figure 38. Network Object for the Non-Secure Interface of the Remote Firewall

The second new network object needed is the remote secure network, where the destination hosts are located. In our case this is the secure network belonging to rs600024e. It is used by a second connection we create later.



Identification

Object Type: Network

Object Name: Network 192.168.100

Description: Secure Netw. of rs600024e

User Information

User Name:

Filter Lifetime: 0

IP Information

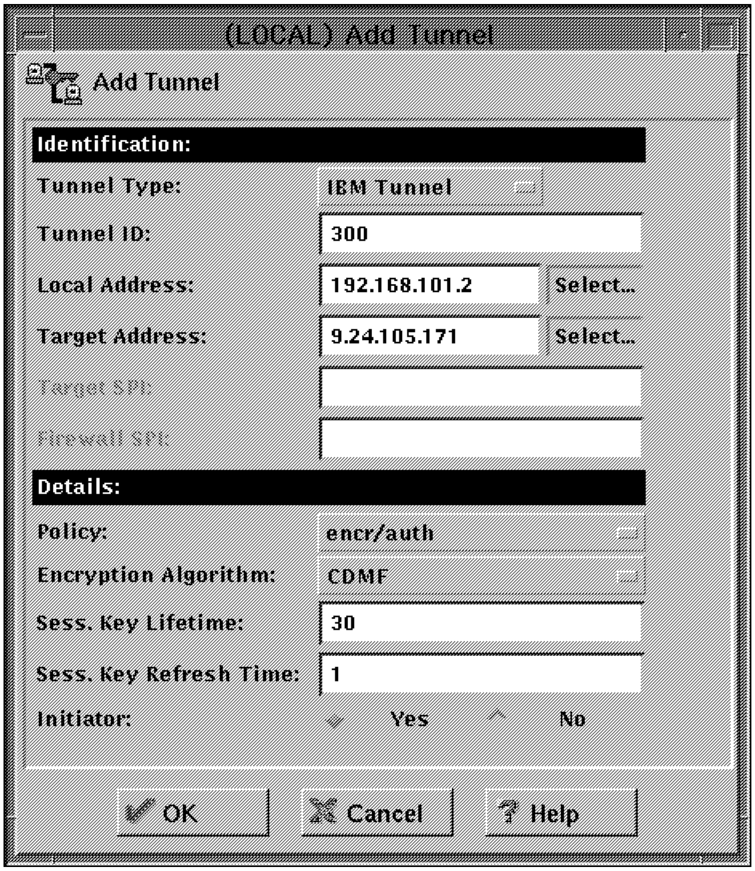
IP Address: 192.168.100.0

Subnet Mask: 255.255.255.0

Figure 39. Network Object for the Secure Remote Network

5.3.3.3 Defining the Tunnel

By clicking on **Virtual Private Network**, **New** and then **Open** we got to the panel for the tunnel definition. For automatic key refresh reasons we chose **IBM Tunnel**. The other fields of the panel were filled in according to the description in 4.1.2.2, “IBM Tunnel” on page 62. The authentication transform is automatically set to keyed MD5. Because of the worldwide availability, we chose **CDMF** as the encryption algorithm. If you are based in the United States, you will want to select the stronger transform DES CBC 8. Based on our discussion in 5.1, “Design Considerations” on page 81 we chose the policy **encr/auth**. For testing purposes we entered a short session key lifetime.



Tunnel Type: IBM Tunnel

Tunnel ID: 300

Local Address: 192.168.101.2

Target Address: 9.24.105.171

Target SPI:

Firewall SPI:

Policy: encr/auth

Encryption Algorithm: CDMF

Sess. Key Lifetime: 30

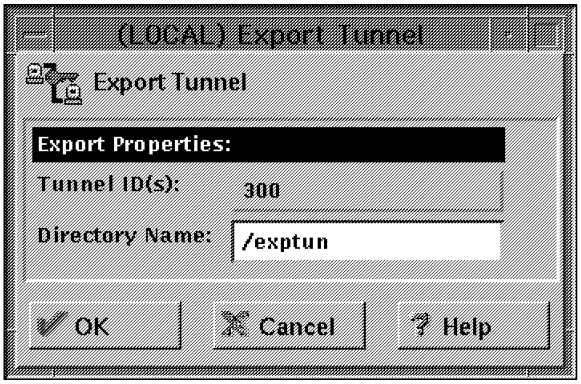
Sess. Key Refresh Time: 1

Initiator: [✓ Yes] No

Figure 40. Tunnel Definition

5.3.3.4 Exporting the Tunnel Definition

The tunnel definition needs to be exported to a directory on your local hard disk in order to be able to transport it to the remote firewall, where the import will take place. The directory specified must already exist. Click on **Export** to get to the corresponding panel. The tunnel ID is the one defined in the step above (and may be chosen randomly as long as it is unique on your system).



Tunnel ID(s): 300

Directory Name: /exptun

Figure 41. Tunnel Export

5.3.3.5 Creating the Packet Filters

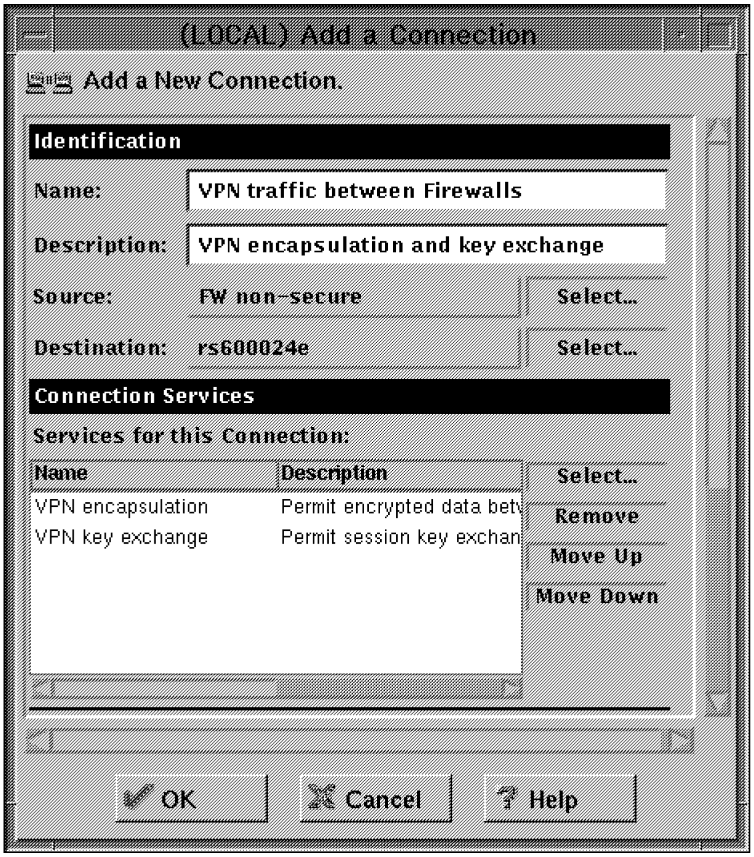
Now as the tunnel definitons are completed, we need to create the packet filter rules for the tunnel traffic. There are two connections involved, and therefore services must be defined for each of them:

1. The first connection handles the tunnel traffic between the firewalls. Therefore the involved objects are the two non-secure interfaces of the firewalls. Actually it does not matter which interface is the source and which is the target object. Add the new connection and include the following predefined services:

   a. VPN encapsulation: This service allows authenticated and encrypted traffic (IPSec AH and ESP packets).

   b. VPN key exchange: This service opens UDP port 4001 for the automatic key exchange.

   An example of a definition for the connection and services described above is shown in Figure 42:



Name: VPN traffic between Firewalls

Description: VPN encapsulation and key exchange

Source: FW non-secure

Destination: rs600024e

Name	Description
VPN encapsulation	Permit encrypted data betv
VPN key exchange	Permit session key exchan

Figure 42. Firewall-to-Firewall Connection

2. The second connection handles the traffic between both secure networks. Those are also network objects. Again source and destination can be exchanged without any negative effects. Use the following predefined services for this connection:

   a. VPN traffic 1/2: This service allows any traffic between both networks through the secure interface.

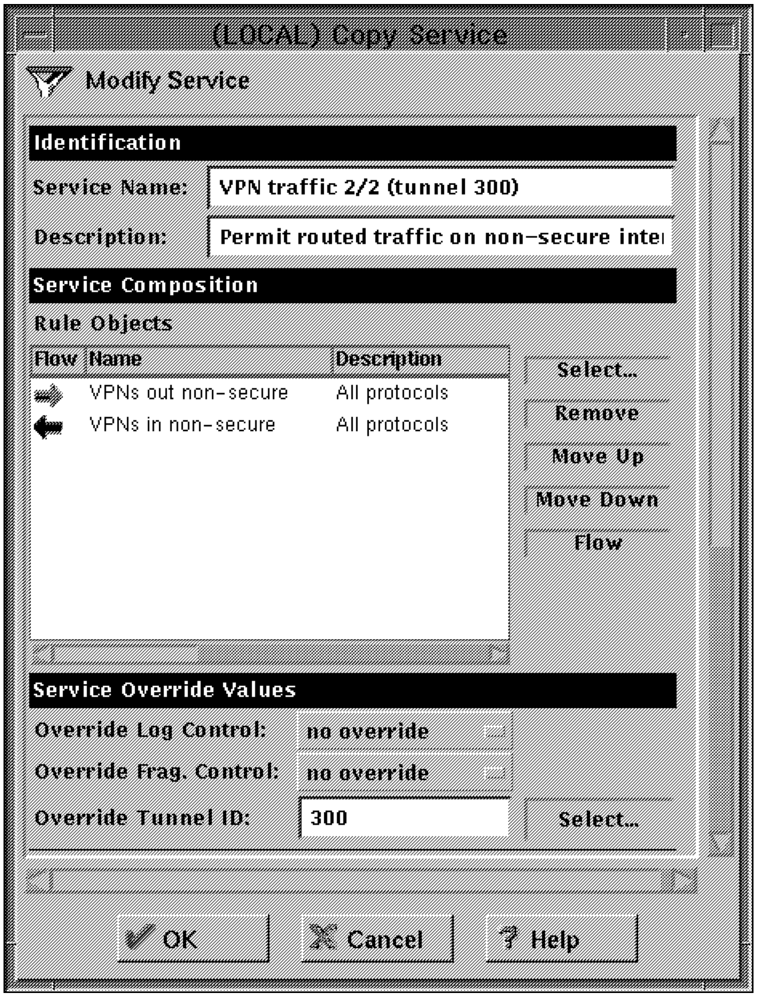
   b. There is also a predefined service for traffic through the non-secure interface of the firewall, but that must be modified. Copy the predefined service VPN traffic 2/2 to a new service called, for instance, VPN traffic

2/2 (tunnel 300). The only field you have to change is Override Tunnel ID. Set it to 300 by using the **Select** button (300 being the tunnel ID specified in the tunnel definition).

The reason why the tunnel ID is needed here is that the software needs filter rules driving the packets through the IPSec kernel before they leave the firewall (as the packets up to now are still in the clear).

We have discussed that this service will drive the packets through the IPSec kernel. Afterwards the now encrypted and/or authenticated packets will be sent directly to the remote firewall over the non-secure interface. (For performance reasons it will not be sent through the filter rules set again.)

An example of a definition for the connection and services described above is shown in Figure 43:



(LOCAL) Copy Service

Modify Service

Identification

Service Name: VPN traffic 2/2 (tunnel 300)

Description: Permit routed traffic on non-secure inte

Service Composition

Rule Objects

Flow	Name	Description
→	VPNs out non-secure	All protocols
←	VPNs in non-secure	All protocols

Service Override Values

Override Log Control: no override

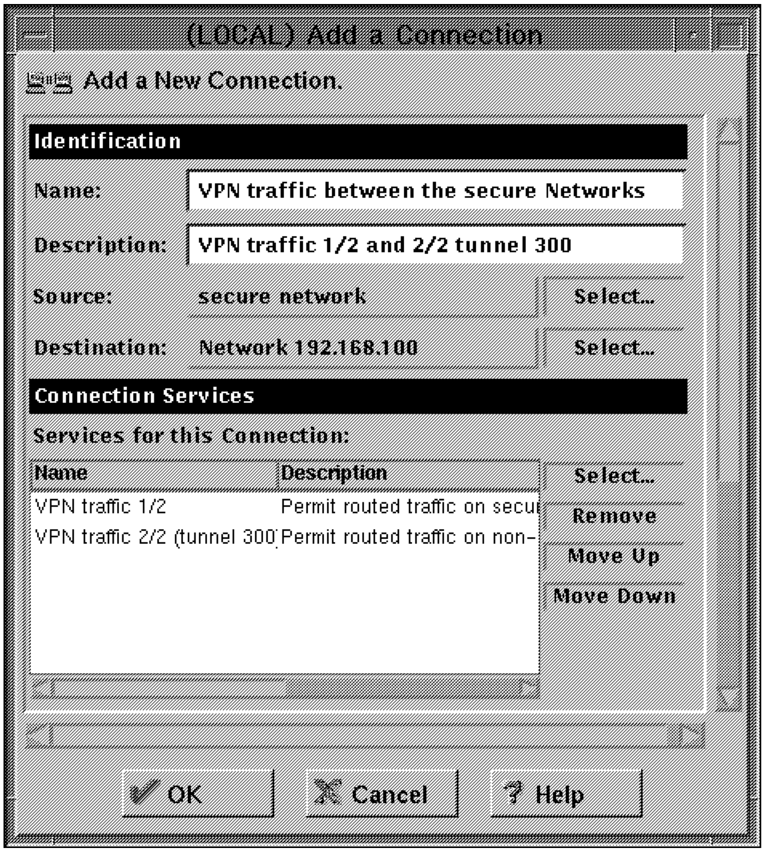
Override Frag. Control: no override

Override Tunnel ID: 300

Figure 43. Service VPN Traffic 2/2 for Tunnel 300

Keep in mind that the firewall uses the security association (SA) type tunnel mode. Therefore the packets that were processed by the IPSec kernel have a new header with a new source and destination address (namely that of the non-secure interface of the two firewalls). See 5.6, “How Does It Work?” on page 104 for more details, but now let′s get back to the configuration procedure.

Now we are able to add the second connection. The objects are the two secure networks and the services to be included are VPN traffic 1/2 and VPN traffic 2/2 (tunnel 300).



(LOCAL) Add a Connection

Add a New Connection.

Identification

Name: VPN traffic between the secure Networks

Description: VPN traffic 1/2 and 2/2 tunnel 300

Source: secure network

Destination: Network 192.168.100

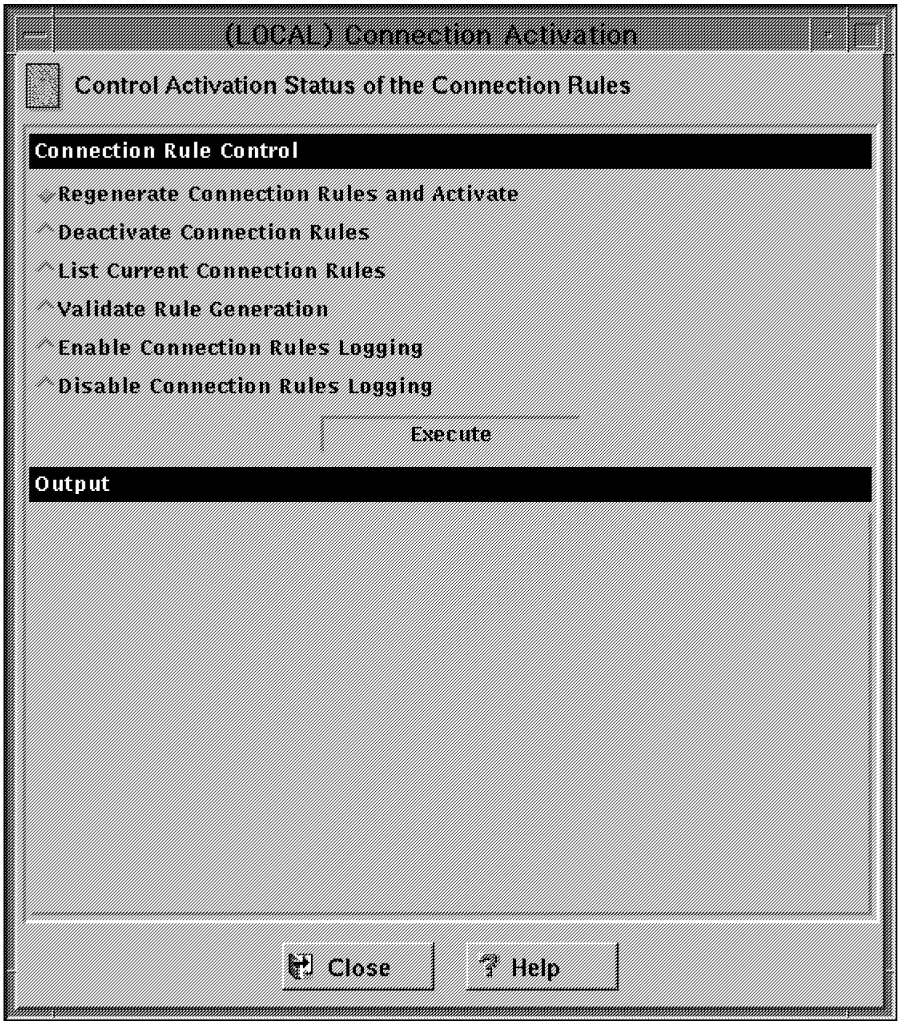
Connection Services

Services for this Connection:

Name	Description
VPN traffic 1/2	Permit routed traffic on secu
VPN traffic 2/2 (tunnel 300	Permit routed traffic on non-

Figure 44. Connection between the Secure Networks

The last step concerning the packet filters is the activation of the new rules.



(LOCAL) Connection Activation

Control Activation Status of the Connection Rules

Connection Rule Control

- ● Regenerate Connection Rules and Activate
- ○ Deactivate Connection Rules
- ○ List Current Connection Rules
- ○ Validate Rule Generation
- ○ Enable Connection Rules Logging
- ○ Disable Connection Rules Logging

Figure 45. Activation of the New Filter Rules

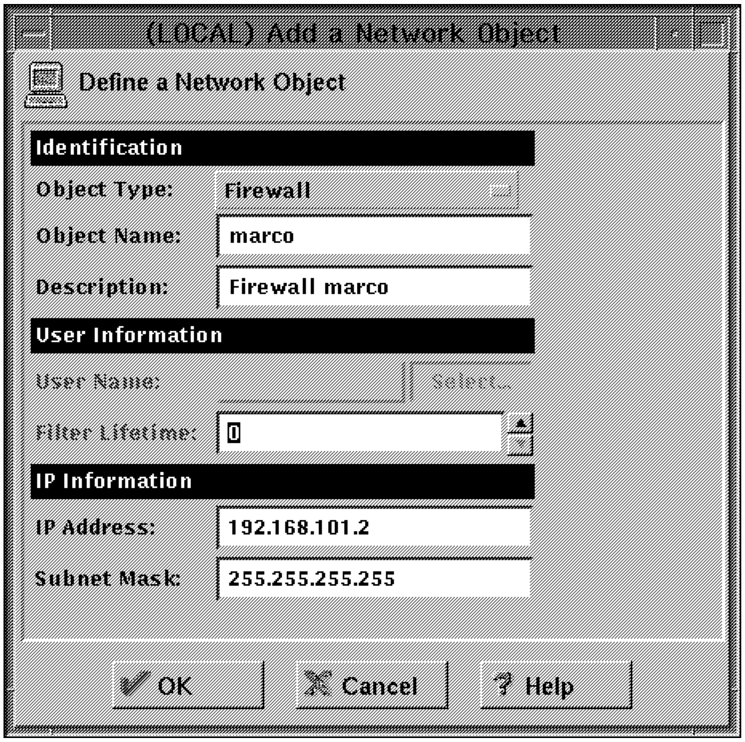
The setup of the first firewall (in our case marco) is now complete.

5.3.3.6 Transporting the Tunnel Definition to the Second Firewall

A simple, but probably not easy step is to get the exported tunnel definition to the second firewall (rs600024e). 6.6, “Manual Key Distribution” on page 131 outlines a few possible solutions for a secure transport of the files. Put the files in a directory of your choice (for instance /imptun) on the second firewall. They will be needed in a later step.

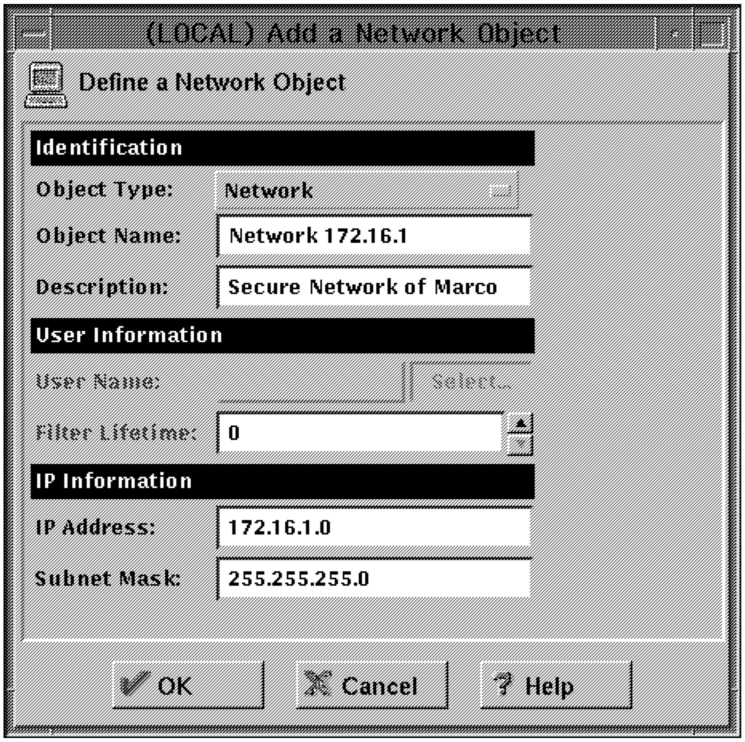
5.3.3.7 Creating the Network Objects in the Second Firewall

On the second firewall (rs600024e) the steps are very similar to the setup we have just discussed. Where the steps do not differ we have shortened the descriptions in order to avoid duplication. Concerning the definition of the network objects we need to define the non-secure interface of the first firewall (marco) and its secure network.



Object Type: Firewall
Object Name: marco
Description: Firewall marco
User Name:
Filter Lifetime: 0
IP Address: 192.168.101.2
Subnet Mask: 255.255.255.255

Figure 46. Network Object for the Non-Secure Interface of the Remote Firewall



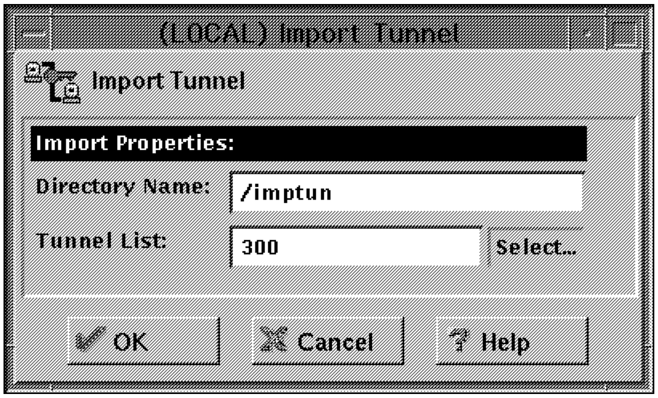
Object Type: Network
Object Name: Network 172.16.1
Description: Secure Network of Marco
User Name:
Filter Lifetime: 0
IP Address: 172.16.1.0
Subnet Mask: 255.255.255.0

Figure 47. Network Object for the Secure Remote Network

In the next step we define the tunnel by importing the definitions from the first firewall.

5.3.3.8 Importing the Tunnel Definition

The import procedure will automatically create the tunnel definition and switch local and remote addresses as local and remote are relative to the specific firewall.



Directory Name: /imptun

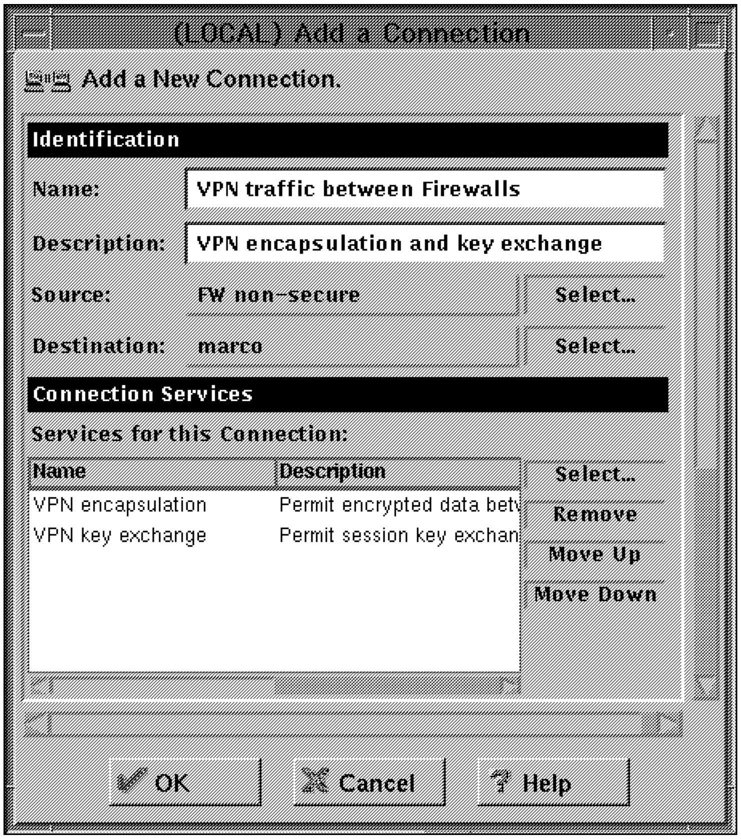
Tunnel List: 300

Figure 48. Tunnel Definition Import

Now the tunnel definition on the second firewall is complete and we need to set up the required packet filters.

5.3.3.9 Creating the Packet Filters

This procedure is exactly the same as for the first firewall (marco); even the objects are the same. For details see 5.3.3.5, "Creating the Packet Filters" on page 93.



Name: VPN traffic between Firewalls

Description: VPN encapsulation and key exchange

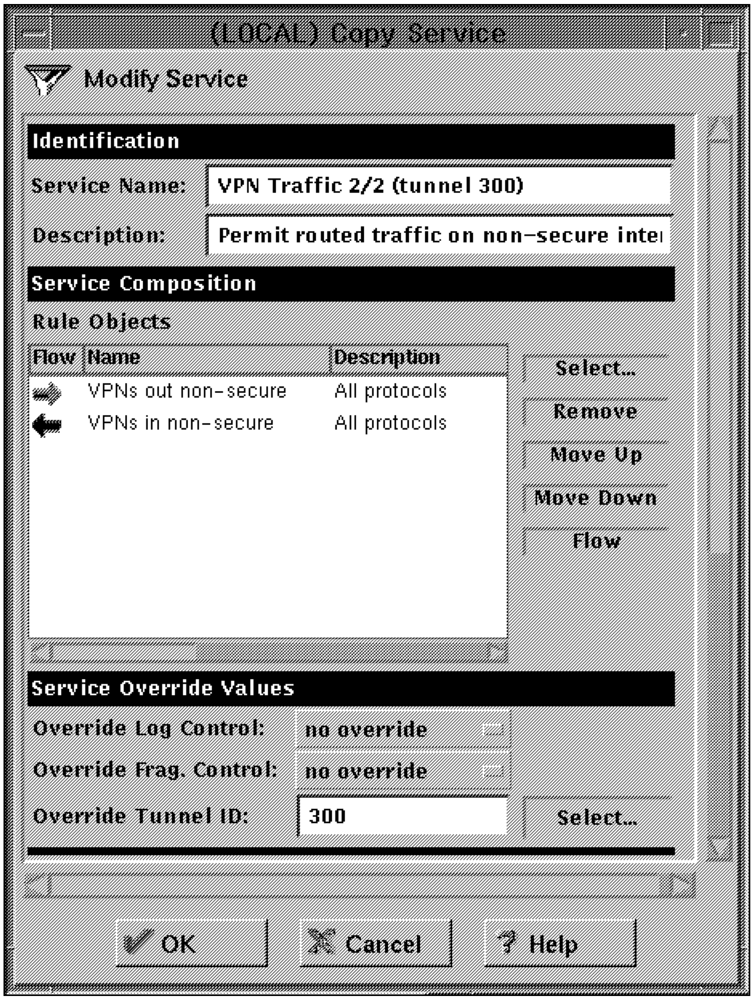
Source: FW non-secure

Destination: marco

Name	Description
VPN encapsulation	Permit encrypted data betv
VPN key exchange	Permit session key exchan

Figure 49. Firewall-to-Firewall Connection

Prior to creating the second connection we have to create the new tunnel service (like we did on the first firewall, see 5.3.3.5, “Creating the Packet Filters” on page 93).



Identification

Service Name: VPN Traffic 2/2 (tunnel 300)

Description: Permit routed traffic on non−secure inte

Service Composition

Rule Objects

Flow	Name	Description
→	VPNs out non−secure	All protocols
←	VPNs in non−secure	All protocols

Service Override Values

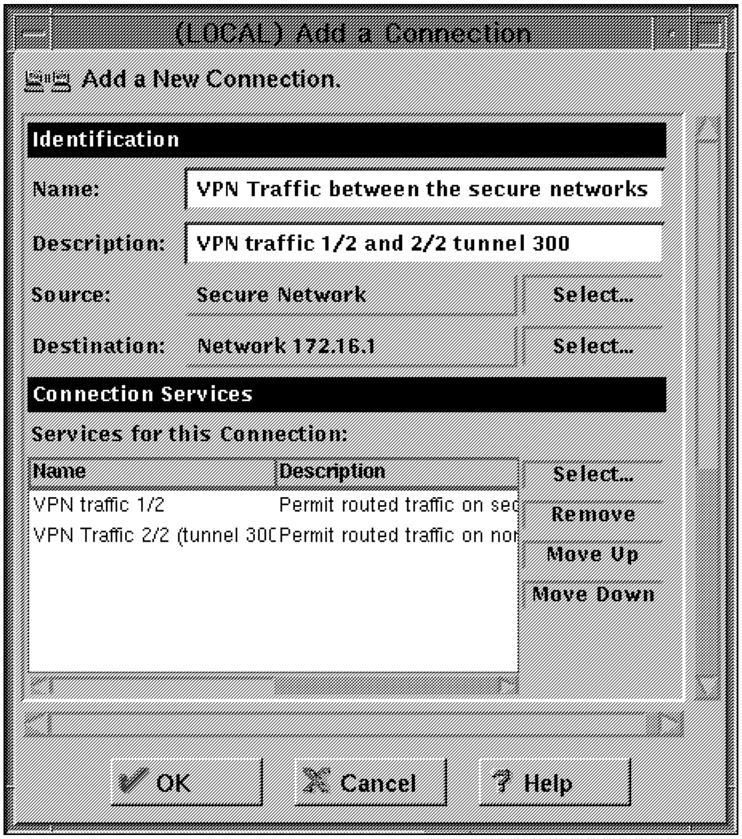
Override Log Control: no override

Override Frag. Control: no override

Override Tunnel ID: 300

Figure 50. Service VPN Traffic 2/2 for Tunnel 300

Now we are ready to create the connection between the secure networks (analogous to 5.3.3.5, “Creating the Packet Filters” on page 93).



(LOCAL) Add a Connection

Add a New Connection.

Identification

Name: VPN Traffic between the secure networks

Description: VPN traffic 1/2 and 2/2 tunnel 300

Source: Secure Network

Destination: Network 172.16.1

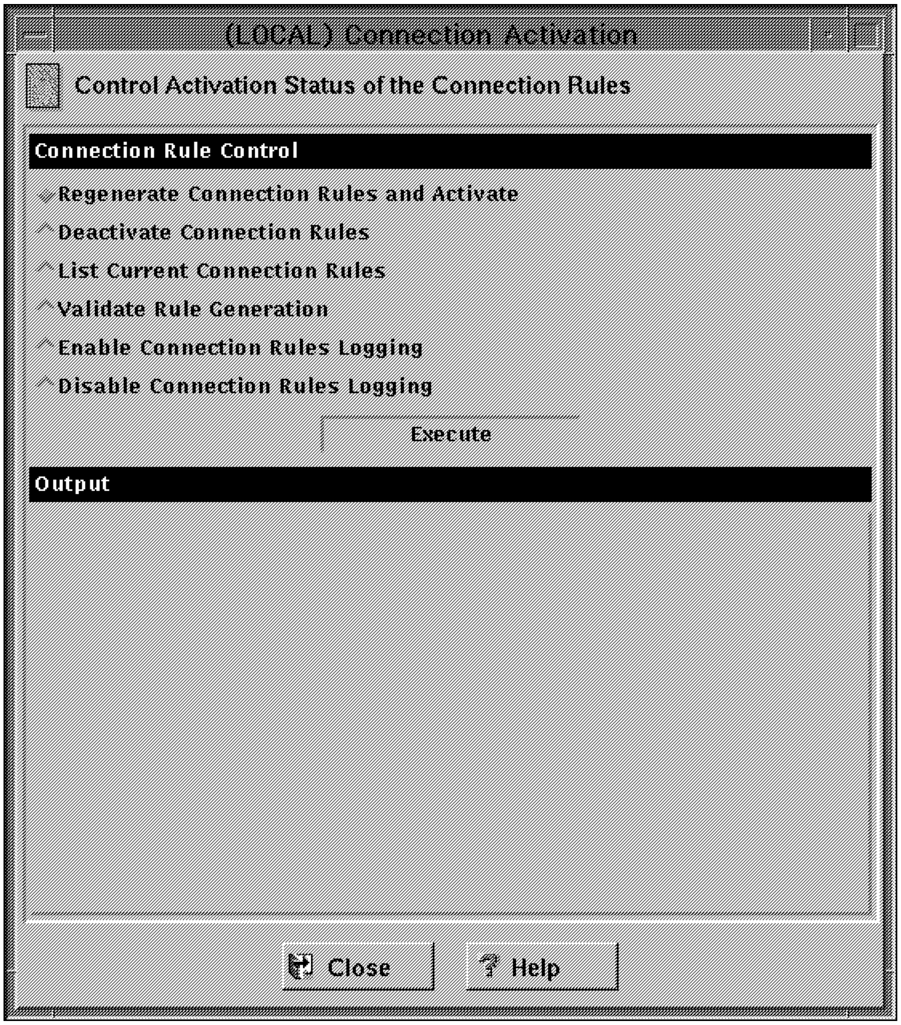
Connection Services

Services for this Connection:

Name	Description
VPN traffic 1/2	Permit routed traffic on sec
VPN Traffic 2/2 (tunnel 30(	Permit routed traffic on nor

Figure 51. Connection between the Secure Networks

The last step in order to complete the packet filter setup is to activate the new rule base.



(LOCAL) Connection Activation
Control Activation Status of the Connection Rules
Connection Rule Control
Execute
Output
Close
Help

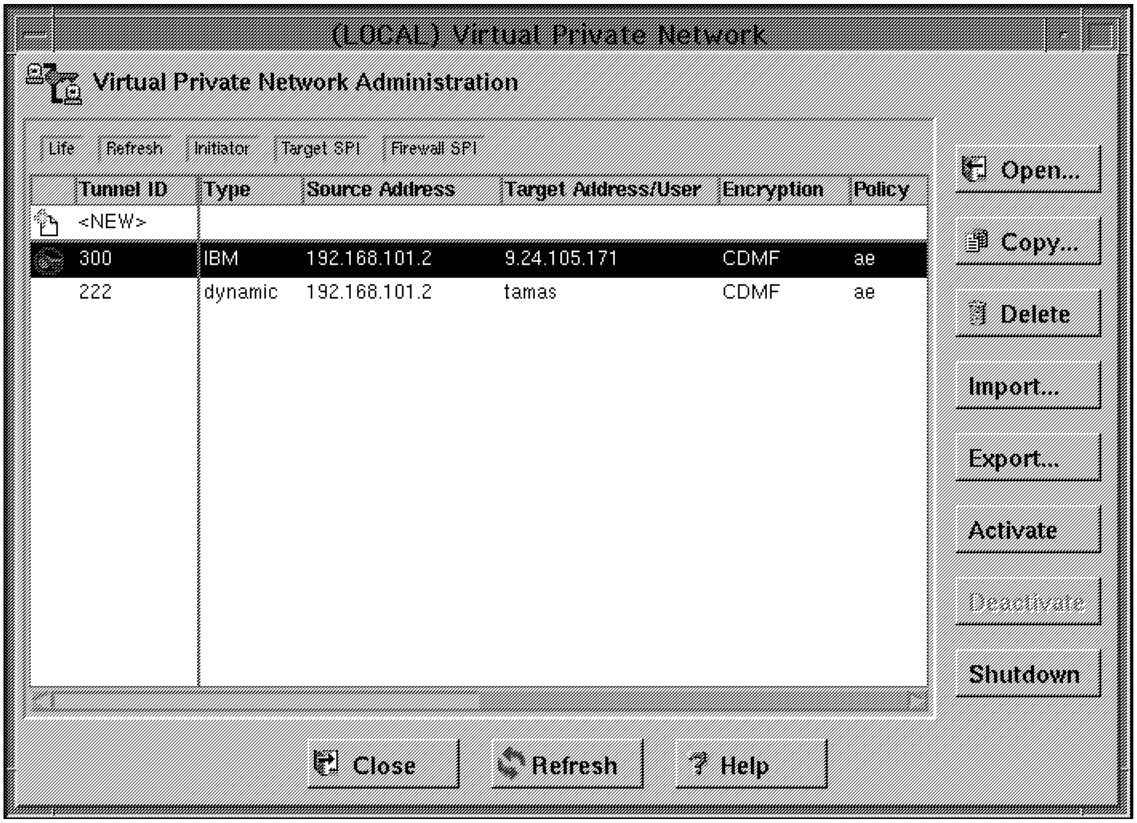
● Regenerate Connection Rules and Activate
○ Deactivate Connection Rules
○ List Current Connection Rules
○ Validate Rule Generation
○ Enable Connection Rules Logging
○ Disable Connection Rules Logging

Figure 52. Activation of the New Filter Rules

This step completes the setup of the second firewall (rs600024e). Finally we need to activate the tunnel at *both* firewalls.

5.3.3.10 Activating the Tunnel at Both Ends

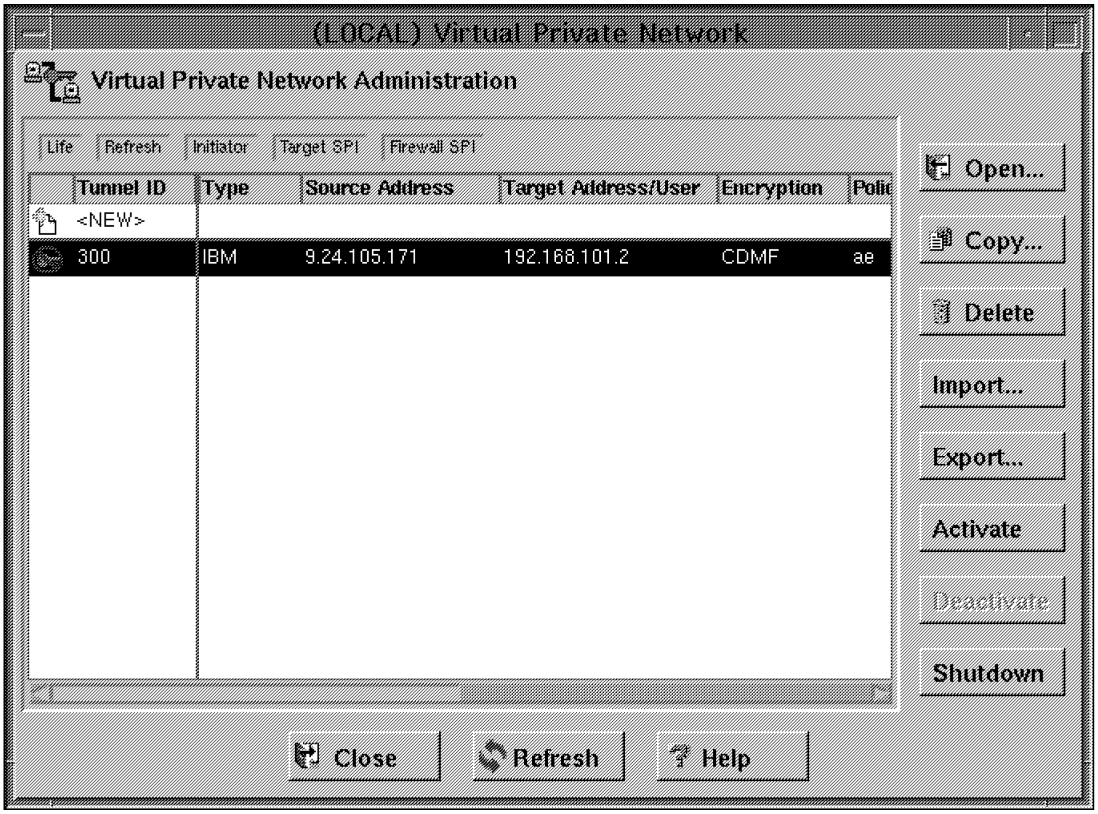
Figure 53 on page 102 shows the activation on the first firewall (marco). Click on **Virtual Private Network**, then on the line containing the tunnel you wish to activate and then on the **Activate** button.



	Tunnel ID	Type	Source Address	Target Address/User	Encryption	Policy
	<NEW>					
	300	IBM	192.168.101.2	9.24.105.171	CDMF	ae
	222	dynamic	192.168.101.2	tamas	CDMF	ae

Figure 53. Tunnel Activation on the First Firewall

Take measures that the tunnel activation at the second firewall takes place (see Figure 54 on page 103). After activation of the tunnel the key symbol in the front of the status line indicates that the tunnel is now active. (The key is crossed out if the tunnel is not active.) A tunnel shown as active does not necessarily mean that the two firewalls are able to communicate. It just means that this side of the tunnel is ready (see also 5.4, “Testing the Tunnel” on page 103).



Tunnel ID	Type	Source Address	Target Address/User	Encryption	Polic
<NEW>					
300	IBM	9.24.105.171	192.168.101.2	CDMF	ae

Figure 54. Tunnel Activation on the Second Firewall

Wait some time (30 seconds should do) for the initialization process to finish. Now your tunnel is ready for traffic.

5.4 Testing the Tunnel

You can easily determine whether the tunnel connection between the firewalls works if you use the admin_test command. (See 8.2.2.1, “The admin_test Command” on page 153 for a description.)

A ping between the firewalls (at least through a tunnel created by using the procedure above) will *not* work because the VPN traffic 2/2 (tunnel 300) filter rules use the route option. (For a detailed explanation see 5.6, “How Does It Work?” on page 104.) This makes sense since you want to allow traffic between the secure networks and not the firewalls. Therefore local firewall traffic through the tunnel will be rejected if you use the default rules. If it works, you have other filter rules allowing a ping between the firewalls. Hence a ping between the firewalls is no way to see whether the tunnel works, but a ping between two hosts in the secure networks is a valid way to test the tunnel.

Note: If your filters are wrong, it could as well be that the admin_test above works but traffic through the tunnel does not. 5.6, “How Does It Work?” on page 104 will help you to understand the way of a packet from one tunnel endpoint to the other.

If there are problems, please refer to Chapter 8, “Troubleshooting Your VPN” on page 151.

5.5 Tunnel Operation

For an IBM tunnel there is actually nothing you have to do in order to keep everything running. It will refresh the session key in time so that the tunnel will stay up.

5.5.1 Activate/Deactivate a Tunnel

We have already seen how to activate a tunnel (Figure 54 on page 103). The deactivation takes place on the same panel. Just click on **Deactivate** instead of Activate. Afterwards any traffic through the tunnel will be rejected until the tunnel is active on both sides again.

5.5.2 When/Why Does an IBM Tunnel Cease Operation?

Under normal conditions an IBM tunnel will never cease operation unless the tunnel is deactivated manually. One other reason for the tunnel to stop is when it tries to refresh the keys and does not succeed (for instance because there is a problem in the firewall-to-firewall connection).

5.6 How Does It Work?

This section follows the packet on the way from one endpoint to the other in order to understand the filter rules necessary for the connection. But before we follow the way a packet travels let us first list the filter rules used for the branch office scenario.

5.6.1 Filter Rules for Tunnel Traffic

For better reading we took out the IP addresses and network masks and replaced them with:

FW1: IP address of firewall 1 (marco)

FW2: IP address of firewall 2 (rs600024)

SN1: Secure network of firewall 1 (marco)

SN2: Secure network of firewall 2 (rs600024)

NM: Network mask of the object

- Firewalls: 255.255.255.255
- Networks: 255.255.255.0 (in our case)

We have also replaced the real tunnel ID with x and numbered the rules (in parentheses, at the end of each rule) so that we can reference them later on throughout the explanations.

Note: We have changed logging to yes to allow easier analyzing (default: Logging=no). The best way to accomplish this is by changing the Log Control field in each of the services to yes.

The two connections (four services) of our configuration example produced the following rules on FW1 (marco):

```
#       Between both firewalls
#        Service : VPN encapsulation
# Description : Permit encrypted data between firewalls
permit FW1 NM FW2 NM ah any 0 any 0 non-secure local both l=y f=y          (1)
permit FW2 NM FW1 NM ah any 0 any 0 non-secure local both l=y f=y          (2)
permit FW1 NM FW2 NM esp any 0 any 0 non-secure local both l=y f=y         (3)
permit FW2 NM FW1 NM esp any 0 any 0 non-secure local both l=y f=y         (4)

#       Between both firewalls
#        Service : VPN key exchange
# Description : Permit session key exchanges for IBM tunnels
permit FW1 NM FW2 NM udp eq 4001 eq 4001 non-secure local both l=y f=y     (5)
permit FW2 NM FW1 NM udp eq 4001 eq 4001 non-secure local both l=y f=y     (6)

#       Between both secure networks
#        Service : VPN traffic 1/2
# Description : Permit routed traffic on secure interface (non-encrypted)
permit SN1 NM SN2 NM all any 0 any 0 secure route inbound l=y f=y          (7)
permit SN2 NM SN1 NM all any 0 any 0 secure route outbound l=y f=y         (8)

#       Between both secure networks
#        Service : VPN traffic 2/2 (tunnel x)
# Description : Permit routed traffic on non-secure interface (encrypted)
permit SN1 NM SN2 NM all any 0 any 0 non-secure route outbound l=y f=y t=x (9)
permit SN2 NM SN1 NM all any 0 any 0 non-secure route inbound l=y f=y t=x (10)
```

Figure 55. Tunnel Filter Rules on FW1 (marco)

Note: Actually the standard description of the last service above is not precise. This service sends traffic to the the IPSec kernel for encryption (and receives it from there), but when passing the rules of this service the packet is in the clear (non-encrypted).

The above filter rules allow for tunnel traffic between any hosts in both secure networks. The following section explains the journey of a packet from Host H1 to Host H2 through a tunnel between FW1 and FW2 (as shown in Figure 56).

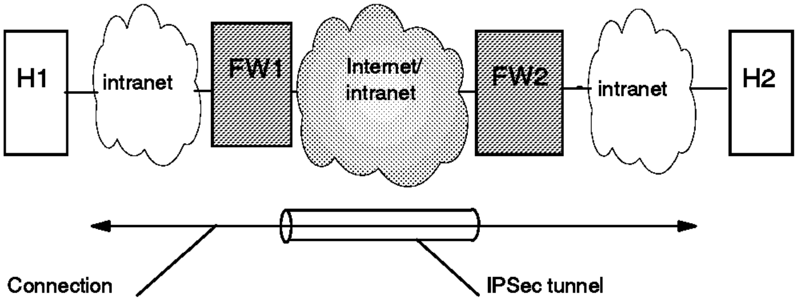


Figure 56. Tunnel between Firewalls FW1 And FW2, Traffic between Hosts H1 And H2

5.6.2 The Flow of a Packet

Now let us take a journey from one tunnel endpoint to the other. We start our travel at the originating host H1 located in the secure network of FW1 (marco) and get to the secure network interface of FW1 via our intranet. That′s where Figure 57 on page 106 starts on the left-hand side.

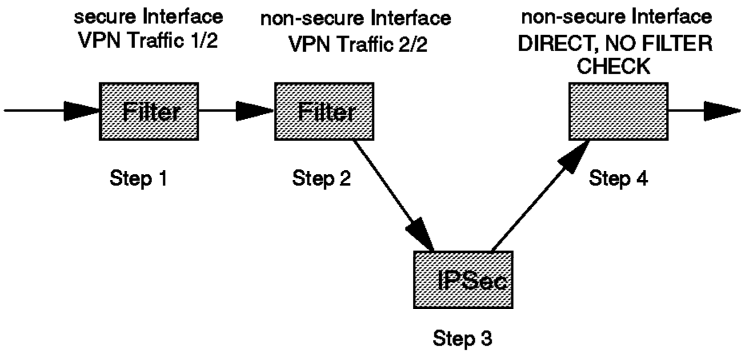


Figure 57. Outbound Packet Flow on Firewall FW1 (marco)

The figure shows the way of the packet from entering the firewall on the secure interface until it leaves the firewall on its way to the Internet. Let′s discuss the details:

Note: The original packet header carries H1 as source address and H2 as destination address. The message starts in the clear.

Step 1
The packet is allowed by rule 7 of Figure 55 on page 105. Because ipforwarding is enabled, the firewall forwards the packet to the non-secure interface of FW1.

Step 2
During the check of the filter rules for the non-secure interface, the definitions of rule 9 are matched. Because this rule defines a non-zero tunnel ID, the packet is sent to the IPSec kernel for processing instead of being sent out to the Internet.

Step 3
The IPSec kernel examines the tunnel ID and processes the packet according to the definitions of the corresponding security association (SA). Our SA defines tunnel mode and the policy encr/auth. Therefore the packet is first encrypted, then authenticated and also gets a new IP header (tunnel mode) with source and destination IP addresses of the firewalls FW1 and FW2.

Step 4
The IPSec kernel will *not* send the packet through the filters again. For performance reasons the packet is instead sent straight through the non-secure interface to the Internet.

The packet then passes the remote firewall FW2 on the way to its final destination host H2. Let us explore the return path on FW1 again (which also explains what happened on FW2).

Figure 58 on page 107 shows the way of the packet from arrival at the non-secure interface of the firewall until leaving the firewall through the secure interface on its way back to host H1.

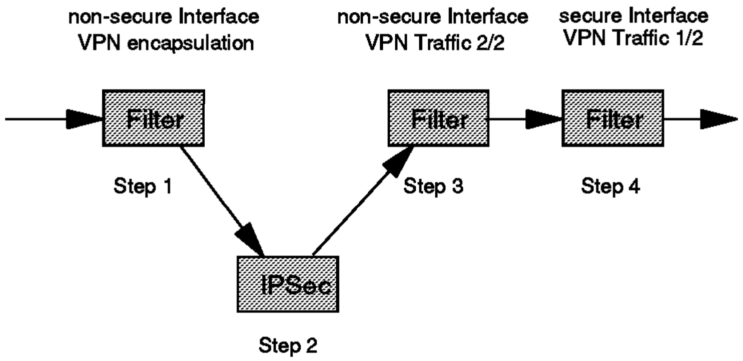


Figure 58. Inbound Packet Flow on Firewall FW1 (marco)

Step 1

The packet gets checked by the filter rule base and matches rule 2 of Figure 55 on page 105 (in our case, because AH was the last protocol applied). Because it carries an IPSec header, the packet will be sent to the IPSec kernel. Up to now the packet is authenticated and encrypted.

Step 2

The IPSec kernel examines the tunnel ID and processes the packet according to the definitions of the corresponding security association(s) (SA). In our case the packet is first authenticated and then decrypted, which means that authentication takes place according to the SPI indicated in the AH header. That header is then stripped off, exposing the ESP header to which the former AH header′s next header field is pointing. Decryption is performed accoring to the SPI indicated in the ESP header, which is then stripped of, exposing the original datagram′s IP header. For better understanding of this process, the whole IP datagram as it is received by the IPSec kernel is shown in Figure 59:

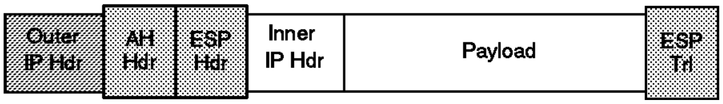


Figure 59. IP Packet Inside a Combined AH-ESP Tunnel

The additional tunnel mode IP header (containing the two firewall addresses) is also removed.

Now the source IP address host H2 and destination IP address host H1 are in the outermost IP header again. The IPSec kernel sends the packet again to the non-secure interface.

Note: If the authentication of the packet fails, it is discarded by the IPSec kernel right away. In that case the firewall logs an ICA1049 message (invalid IPSec package) that you can scan for to find out if someone tries to mount a denial-of-service attack.

Step 3

The packet is checked by the filter rule base and matches rule 10. Routing is enabled; therefore the packet is forwarded to the secure interface.

Step 4

On this interface the packet matches rule 8 and is sent to the intranet where it finally arrives back at host H1.

The additional rules 5 and 6 (udp 4001) are necessary to enable the automatic key refresh. They are not needed if you use manual or dynamic tunnels. The default rules allow AH and ESP packets between the firewalls. Actually in our case we could as well delete rules 3 and 4. We use ESP, but it is encapsulated within AH; therefore on the Internet (and also in every trace) there will be only AH packets.

5.7 Variations of the Branch Office Scenario

Up to now we have discussed the basic branch office scenario. It may vary in different ways. Below we picked out three cases.

5.7.1 Allow Only Firewall-to-Firewall Traffic

Sometimes it may be desirable to create a tunnel between two firewalls and permit only traffic between the firewalls themselves (for instance, if you would like to be able to administer one firewall from another remote firewall).

The tunnel definition will be the same as described above, but you will have to change the packet filters as shown below:

```
#       Between both firewalls
#        Service : VPN encapsulation
# Description : Permit encrypted data between firewalls
permit FW1 NM FW2 NM ah any 0 any 0 non-secure local both l=y f=y         (1)
permit FW2 NM FW1 NM ah any 0 any 0 non-secure local both l=y f=y         (2)
permit FW1 NM FW2 NM esp any 0 any 0 non-secure local both l=y f=y        (3)
permit FW2 NM FW1 NM esp any 0 any 0 non-secure local both l=y f=y        (4)

#       Between both firewalls
#        Service : VPN key exchange
# Description : Permit session key exchanges for IBM tunnels
permit FW1 NM FW2 NM udp eq 4001 eq 4001 non-secure local both l=y f=y    (5)
permit FW2 NM FW1 NM udp eq 4001 eq 4001 non-secure local both l=y f=y    (6)

#       Between both firewalls
#        Service : VPN traffic (tunnel x)
# Description : Permit local traffic on non-secure interface (encrypted)
permit FW1 NM FW2 NM all any 0 any 0 non-secure local outbound l=y f=y t=x (7)
permit FW2 NM FW1 NM all any 0 any 0 non-secure local inbound l=y f=y t=x  (8)
```

Figure 60. Filter Rules for Firewall-to-Firewall Traffic

This will allow any traffic between both firewalls through tunnel x. Rules 1 to 6 stayed the same; we only modified rules 7 and 8. We do not need rules 9 and 10 anymore, because the firewall tunnel traffic takes only place at the non-secure interface. Apart from the packet filter change above the whole firewall setup stays the same. Because the tunnel traffic is restricted to both firewalls, you are now able to put all tunnel related filter rules into one connection (source object FW1, destination object FW2). Don′t forget that those definitions are required on both firewalls.

5.7.2 Allow Only Certain Kinds of Traffic

Within the base branch office scenario there are no restrictions concerning the type of traffic flowing through the tunnel. You may wish to restrict the traffic between the two secure networks to certain activities. You only need to change the connection handling the tunnel traffic between the two secure networks on both firewalls in order to accomplish restrictions. In the example below we restricted the tunnel traffic to the TCP protocol. Everything else (such as ICMP, UDP and so on) is rejected. The other tunnel filter rules (1-6) stay the same.

```
#       Between both firewalls
#        Service : VPN encapsulation
# Description : Permit encrypted data between firewalls
permit FW1 NM FW2 NM ah any 0 any 0 non-secure local both l=y f=y          (1)
permit FW2 NM FW1 NM ah any 0 any 0 non-secure local both l=y f=y          (2)
permit FW1 NM FW2 NM esp any 0 any 0 non-secure local both l=y f=y         (3)
permit FW2 NM FW1 NM esp any 0 any 0 non-secure local both l=y f=y         (4)

#       Between both firewalls
#        Service : VPN key exchange
# Description : Permit session key exchanges for IBM tunnels
permit FW1 NM FW2 NM udp eq 4001 eq 4001 non-secure local both l=y f=y     (5)
permit FW2 NM FW1 NM udp eq 4001 eq 4001 non-secure local both l=y f=y     (6)

#       Between both secure networks
#        Service : VPN traffic 1/2
# Description : Permit routed traffic on secure interface (non-encrypted)
permit SN1 NM SN2 NM tcp any 0 any 0 secure route inbound l=y f=y          (7)
permit SN2 NM SN1 NM tcp any 0 any 0 secure route outbound l=y f=y         (8)

#        Service : VPN traffic 2/2 (tunnel x)
# Description : Permit routed traffic on non-secure interface (encrypted)
permit SN1 NM SN2 NM tcp any 0 any 0 non-secure route outbound l=y f=y t=x (9)
permit SN2 NM SN1 NM tcp any 0 any 0 non-secure route inbound l=y f=y t=x (10)
```

Figure 61. Filter Rules for Special Traffic

5.7.3 Allow Only Traffic between Specific Hosts

Another variation of the branch office scenario is that you still trust your secure networks but you only want to allow specific hosts to use the tunnel to connect to the remote secure network.

The only difference to the base setup described is again in the filter rule base. For the connection between the secure networks (rules 7 to 10 in 5.6, "How Does It Work?" on page 104) you have to exchange the network objects to the hosts you want to grant tunnel usage. Of course it is possible to put multiple hosts into network object groups and use the group objects instead. The example below assumes that you only want to allow the hosts H1 and H2 access through the tunnel. For this example we assume that you have created two network objects called H1 and H2. Remember the changes need to be done on both firewalls.

```
#       Between both firewalls
#        Service : VPN encapsulation
# Description : Permit encrypted data between firewalls
permit FW1 NM FW2 NM ah any 0 any 0 non-secure local both l=y f=y          (1)
permit FW2 NM FW1 NM ah any 0 any 0 non-secure local both l=y f=y          (2)
permit FW1 NM FW2 NM esp any 0 any 0 non-secure local both l=y f=y         (3)
permit FW2 NM FW1 NM esp any 0 any 0 non-secure local both l=y f=y         (4)

#       Between both firewalls
#        Service : VPN key exchange
# Description : Permit session key exchanges for IBM tunnels
permit FW1 NM FW2 NM udp eq 4001 eq 4001 non-secure local both l=y f=y     (5)
permit FW2 NM FW1 NM udp eq 4001 eq 4001 non-secure local both l=y f=y     (6)

#       Between H1 and H2
#        Service : VPN traffic 1/2
# Description : Permit routed traffic on secure interface (non-encrypted)
permit H1 NM H2 NM all any 0 any 0 secure route inbound l=y f=y            (7)
permit H2 NM H1 NM all any 0 any 0 secure route outbound l=y f=y           (8)

#       Between H1 and H2
#        Service : VPN traffic 2/2 (tunnel x)
# Description : Permit routed traffic on non-secure interface (encrypted)
permit H1 NM H2 NM all any 0 any 0 non-secure route outbound l=y f=y t=x   (9)
permit H2 NM H1 NM all any 0 any 0 non-secure route inbound l=y f=y t=x   (10)
```

Figure 62. Filter Rules for Specific Host-to-Host Traffic

This last example leads us to the next scenario, in which we connect two hosts within the secure networks again, but this time we do not trust the secure networks anymore.

Chapter 6. Business Partner/Supplier Network Scenario

This chapter explores an extension of the branch office scenario, the secure connection between two hosts belonging to untrusted intranets over the public Internet. The focus in this scenario expands from protecting your intranets from outside attacks to protecting single hosts within the intranets also from inside attacks.

Consider a situation where a manufacturing company needs to communicate regularly with its suppliers, for example to facilitate just-in-time delivery of parts, to settle invoices among themselves, or for any number of other reasons. There are two issues to consider:

- Access Control: While it may be a business necessity for supplier A to have access to some of company X′s internal resources (such as databases), there will also be valid business reasons to prevent the supplier from having access to all of company X′s databases.
- Data Confidentiality: Clearly the data should be hidden from general view while it is in transit over the public Internet. But there may be even more stringent requirements. Company X may consider its own intranet to be trusted, but its suppliers may not. For example, a supplier may want to insure that its sensitive data, while traveling through company X′s intranet, is hidden until it reaches its final destination. For example, the supplier may be worried that an unscrupulous eavesdropper inside company X may try to intercept the data and sell it to a competitor. And company X may have the same concerns about its data as it travels though the supplier′s intranet. Thus, it will not be unusual for each party to treat the other′s intranet as untrusted.

6.1 Design Considerations

This scenario is an extension of the multiple branch office scenario. Here we have multiple supplier intranets that need to access a common corporate network over the Internet. Each supplier is allowed access to only a limited set of destinations within the corporate network. Even though traffic from the different suppliers flows over common data links in both the public Internet and in the destination intranet, the VPN must be constructed to guarantee that no traffic from a given supplier will be visible to any other supplier or to any system other than its intended destination.

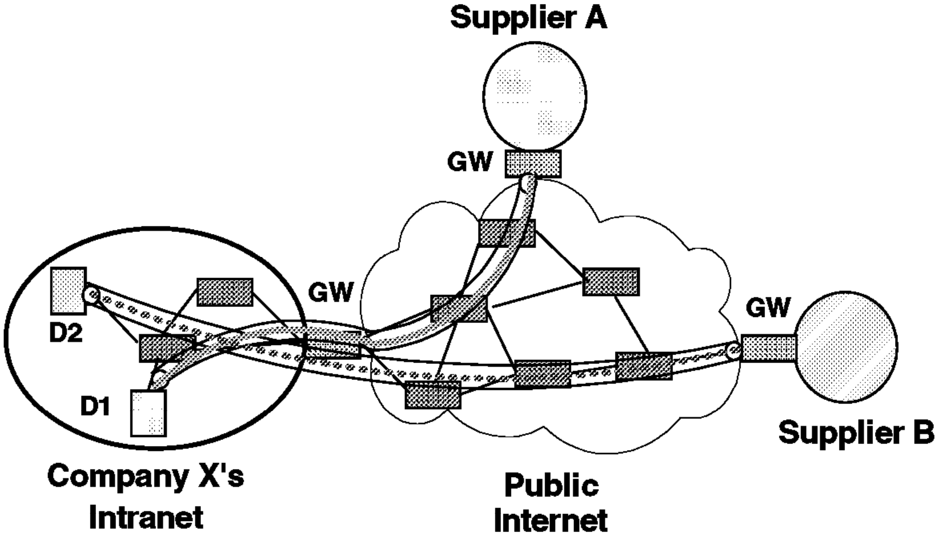


Figure 63. A Typical Supplier Configuration

Figure 63 illustrates how the two data paths, represented by the dashed and solid lines inside the VPN tunnels, can flow through several common boxes. In this example, supplier A can talk only to destination D1 and supplier B can talk only to destination D2. Traffic from suppliers A and B can be intermixed both within the Internet and within company X′s intranet.

IPSec provides a secure solution in this environment, but it will be more complex than for the branch office scenario outlined in Chapter 5, “Branch Office Connection Scenario” on page 81. The extra complexity arises from the following factors:

- There can be multiple suppliers who need to communicate with the manufacturer. Hence, it may be necessary to insure that supplier A can never see any other supplier′s data in cleartext form, either in the Internet or in the manufacturer′s intranet.
- If the manufacturer and the suppliers, or some subset of them, use private addressing in their respective intranets, then it is possible that *routing collisions* can occur if the same private address has been assigned to multiple hosts. To avoid this possibility, the members of the VPN must either use public IP addresses in their intranets, coordinate the assignment of private IP addresses among the systems participating in the VPN, or adopt some sort of Network Address Translation strategy.
- Because security coverage extends from host-to-host (client-to-server) rather than just from gateway-to-gateway, there will be many more security associations to be negotiated, and many more keys to be securely distributed and refreshed, as compared to the branch office scenario. Hence, the automated secure functions of ISAKMP/Oakley will become even more important.
- Because security coverage extends from host-to-host, IPSec functions will need to be supported in clients, servers, and firewalls.

6.1.1 Authenticating and Encrypting Supplier Traffic

As shown in Figure 64, the VPN gateway that guards the entry to company X′s intranet must accept traffic from both supplier A and supplier B. This can be accomplished by using IPSec′s AH protocol. There will be one tunnel between the firewall of company X and supplier A and another between the firewall or router of company X and supplier B. The AH protocol will be used in tunnel mode, providing cryptographically strong access control. Therefore systems in supplier intranet A can communicate with destination D1, and systems in supplier intranet B can communicate with destination D2.

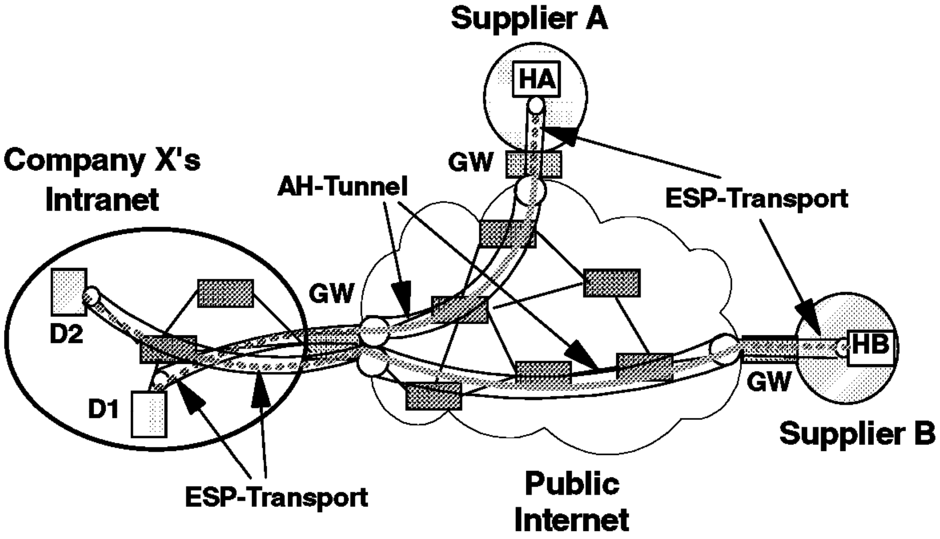


Figure 64. A Typical Supplier Configuration

But as we have noted, there is a need for even finer-grained authentication, namely, each source to its intended destination. For example, in Figure 63 on page 112, we need to assure that destination D1 will accept traffic only from host HA and not from host HB. To achieve data confidentiality, we will use end-to-end encryption between each host and its intended destination server (for example, from host HA to destination D1). IPSec protocols provide the means to accomplish this by using *bundled security associations* (SA bundles), which make use of both tunnel and transport modes of operation simultaneously.

To handle the host-to-host authentication and encryption requirements, we will establish a security association (SA) between each client machine and its server. The protocol will be ESP with authentication, and the type of SA will be transport mode, since this is an end-to-end security association.

Next, we establish a different security association between the gateways that protect company X′s intranet and the supplier′s intranet. This SA applies over only part of the complete path, so it will use the AH protocol in tunnel mode. Because of tunnel mode, the packet will have the gateway′s IP addresses in the ″outer″ IP header. Therefore also private addresses could be used on the intranets. (See 6.1.2, “Addressing Issues” on page 114 for details.) Between firewalls or routers, ESP security association will be nested inside the AH security association. Figure 65 on page 114 illustrates the structure of the datagram that flows between firewalls or routers. An inner datagram is nested inside an outer datagram to support two distinct bundled security associations: client-to-server and gateway-to-gateway.

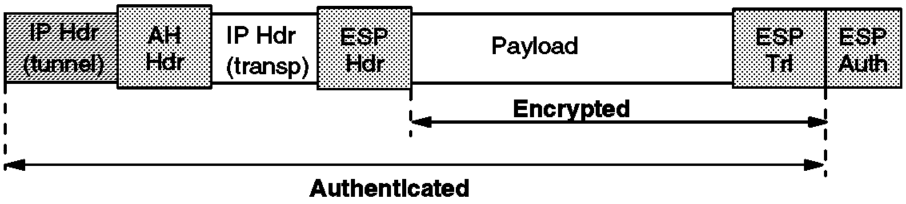


Figure 65. A Typical Supplier Scenario Datagram

Note that IPSec protocols enforce two levels of authentication: firewall-to-firewall and client-to-server. The firewall-to-firewall authentication prevents denial-of-service attacks by making sure that only traffic from legitimate suppliers can enter company X′s intranet; the host-to-host authentication assures that the destination will accept traffic only from its intended partner machines.

This considerably exacerbates scaling issues. Unlike the branch office case where security associations were established only between VPN firewalls or routers, it is now necessary to establish two additional security associations per client. Each security association will require its own set of cryptographic keys. This scenario illustrates the need for automated ISAKMP-based methods, both for negotiating multiple bundled security associations and for distributing the associated keys.

6.1.2 Addressing Issues

In Figure 64 on page 113 there are tunnels between supplier A and company X, and also between supplier B and company X, but there is no tunnel between supplier A and supplier B. For routing purposes, supplier A and company X will run a mutually acceptable routing protocol over their tunnel, and company X and supplier B will also independently run their own routing protocol. Because each tunnel has its own security association, routing data for supplier A can be kept secret from supplier B, and vice versa. As in the case of the Branch Office Interconnection scenario, each security association will use IPSec′s ESP protocol to both encrypt and authenticate the routing updates.

Unlike the Branch Office case, where we could assume that a consistent addressing plan had been applied across all the company′s intranets, in this configuration it is very likely that company X and each of its suppliers have administered their own addressing plan independently of one another. For example, it would be possible that supplier A and supplier B both used private (globally ambiguous) IP addresses in their networks, and it would be possible for some or all of their addresses to overlap. In this case, conventional IP routing protocols will not be able to resolve these ambiguities. Hence, we will make the assumption that the IP addresses of all systems, both in the corporate intranet and in the suppliers′ intranets, have been assigned so that they are non-overlapping. That is, we will assume that when private IP addresses are used, there will be coordination between the communicating intranets.

Note: As mentioned in 1.3.3.1, "Network Address Translation" on page 14, NAT will not help in this case because it will change IP address information which will cause IPSec authentication to fail. In fact, since we need to build end-to-end IPSec tunnels in this scenario, NAT will prohibit the proper setup of security associations alltogether.

6.1.3 Packet Filtering and Proxies

In this configuration, we have seen that there is a requirement for end-to-end encryption. This can cause problems for conventional packet filtering techniques, since the TCP header is part of the encrypted payload field and is no longer visible to the VPN firewalls or routers. Another area that needs to be addressed is the nesting of IPSec protocols. This means that the VPN firewall or router must be able to handle IP packets where the Next protocol field might indicate AH or ESP. It may also mean that packet filters will need to operate on both ″inner″ and ″outer″ IP address information, in cases where tunnel mode is used.

This area needs more study. The effectiveness of packet filtering will be significantly reduced, since unencrypted upper layer data is no longer available for examination by the VPN firewall or router. As the cryptographic techniques become used more widely for end-to-end protection, more and more access control decisions in a firewall will be handled via the AH protocol, and conventional packet filtering will become less and less useful. However, for traditional non-VPN traffic such as everyday World Wide Web access or news, packet filtering will still play its usual role. At the final destination host, where cleartext data is once again available, packet filtering will also continue to play a useful role for providing finer-grained level of access control within the destination host itself.

6.1.4 Summary: Inter-Company Interconnection

This application of IPSec uses the public Internet to connect a company and its suppliers. It requires upgrades to existing client and server machines, since they must now support the IPSec protocol suite. It requires enhancements to conventional packet filtering techniques, because some headers from upper layer protocols may no longer be decryptable at the VPN firewall or router. And finally, it makes use of IPSec′s nesting capabilities. The major elements of complexity, compared to the branch office case, are summarized below:

- Client machines (hosts and servers) must support IPSec′s ESP protocol, both for encryption and for authentication.
- The number of machines that need to participate in the IPSec protocols has increased significantly. Security associations will need to be set up both end-to-end and gateway-to-gateway.
- For very small configurations, manual key distribution and manual configuration of security associations may be possible, but for any medium to large-sized configuration, support for ISAKMP/Oakley in clients, servers, and VPN-firewalls will rapidly become a necessity.
- New packet filtering rules will need to be developed to accommodate: a) encrypted upper layer payloads, and b) pairs of inner and outer cleartext headers that arise when IPSec protocols are nested within one another. It remains to be seen if firewall or router filtering rules in the presence of end-to-end encryption will continue to serve a useful purpose. In the long term, filtering′s importance will probably diminish as cryptography-based access control techniques become more widely used.

6.2 Scenario Setup

We built the test network shown in Figure 66.

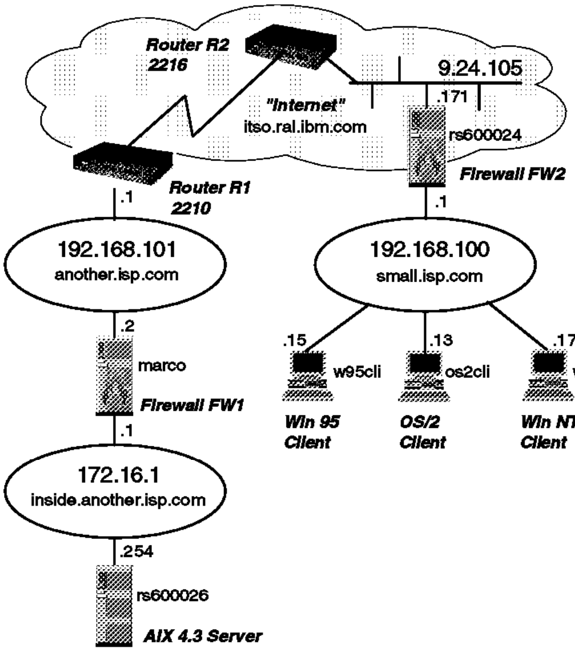


Figure 66. Business Partner/Supplier Scenario - Test Network

The Internet is simulated by the ITSO internal itso.ral.ibm.com subnet, to which we attached our private subnets representing the different sites.

The subnet inside.another.isp.com corresponds to company X′s site, which is attached to the Internet via the router R1. The firewall FW1 (marco) protects the internal subnet inside.another.isp.com. In this subnet reside the company′s servers that run business critical applications. These must be made accessible to the business partners/suppliers. In our configuration, the application server is an AIX V4.3 system (rs600026). The server is IPSec-enabled and we use its IPSec capabilities for the nested tunnel. We access it from the supplier network via a telnet connection.

The supplier subnet small.isp.com connects directly to the internet via the firewall FW2 (rs600024). In real life there usually would be a router as well, but from the VPN point of view it is transparent. The clients represent the current choice of most common client operating systems: Windows 95, Windows NT, and OS/2. For our scenario we installed eNetwork Communications Suite V1.1 on the Windows 95 system (w95cli) and used the IPSec capabilities of the eNetwork Communications Suite to build the end-to-end tunnel to the AIX V4.3 server.

Note: Unfortunately the eNetwork Communications Suite software does not support authentication within the ESP packet, which would be the ideal solution (see the scenario described in 6.1.1, “Authenticating and Encrypting Supplier Traffic” on page 113). Therefore we used encryption only for the nested tunnel, but we had to choose the old header format, because it is the only choice with the eNetwork Communications Suite. If you use products that support the new headers (such as, for instance AIX V4.3), we highly recommend to use this option for the nested tunnel.

The tunnel configuration between the two firewalls is almost the same as in the Branch Office scenario. One difference is that we now use a policy of authentication only, because the nested tunnel handles the encryption issue (as shown in Figure 64 on page 113). The other difference lies in the network objects used. Instead of the two secure networks we now use objects for the two hosts mentioned above, because we want to restrict tunnel traffic to the two hosts.

Both firewall machines run AIX V4.2.1 and eNetwork Firewall for AIX V3.1.1.5. We have also tested the same scenario successfully using AIX V4.3.0 and eNetwork Firewall for AIX V3.2.1 on both machines.

Figure 67 outlines how we configured the tunnels for this scenario.

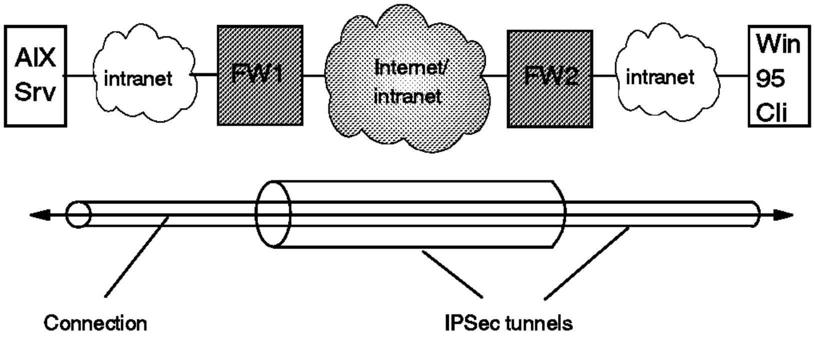


Figure 67. Business Partner/Supplier Scenario - Nested Tunnels to Provide End-to-End Security

The following sections guide you through the configuration of the tunnel between the firewalls and the nested tunnel between the AIX V4.3 and the Windows 95 system.

6.3 Configuration of the Tunnel between the Firewalls

In order to avoid unnecessary duplication we only discuss the differences to the procedures described in 5.3, “Configuration of the Firewalls” on page 87. It is absolutely necessary that you understand the Branch Office scenario setup discussed in Chapter 5, “Branch Office Connection Scenario” on page 81. Otherwise you will not be able to successfully set up this scenario. Please follow the steps described there and replace the following (in the setup of *both* firewalls):

1. Network objects:

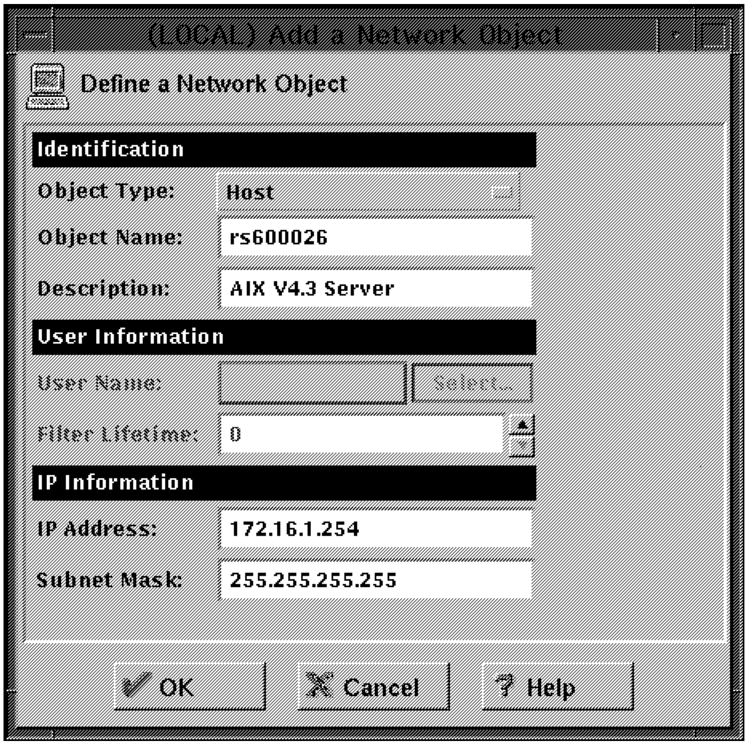
   Secure network of FW1: Replaced by AIX V4.3 server (rs600026) object

   Secure network of FW2: Replaced by Windows 95 system (w95cli) object

2. Tunnel definition: Choose the policy **authentication only**.

Apart from these changes everything stays the same. Attached you find the screen shots of the three actions from above. (We took the images from marco. The images for rs600024 are the same.)

We start by defining the network object for the AIX V4.3 system (see Figure 68 on page 118).



Define a Network Object

Identification

Object Type: Host

Object Name: rs600026

Description: AIX V4.3 Server

User Information

User Name:

Filter Lifetime: 0

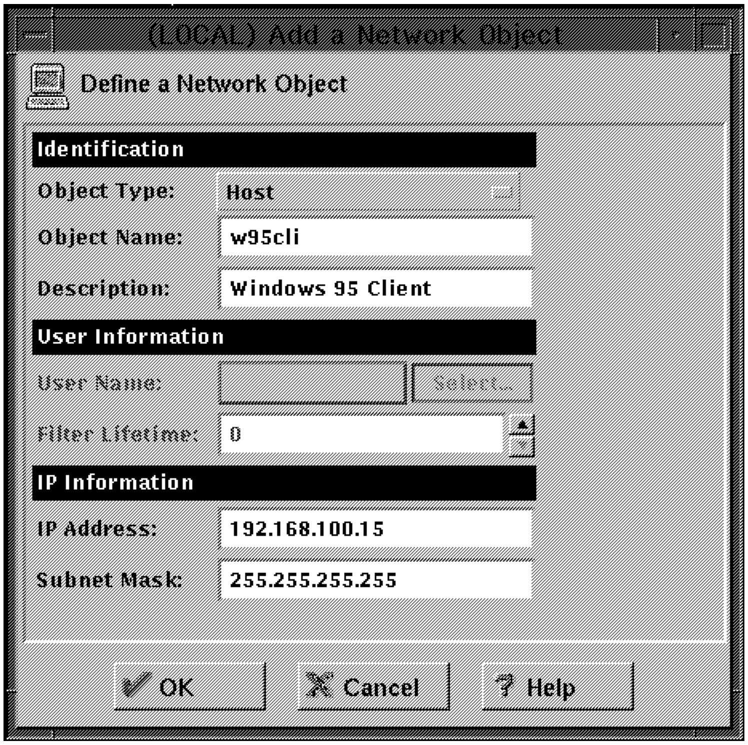
IP Information

IP Address: 172.16.1.254

Subnet Mask: 255.255.255.255

Figure 68. Network Object for the AIX V4.3 Server

In the next step we create the object for the Windows 95 system (see Figure 69).



Define a Network Object

Identification

Object Type: Host

Object Name: w95cli

Description: Windows 95 Client

User Information

User Name:

Filter Lifetime: 0

IP Information

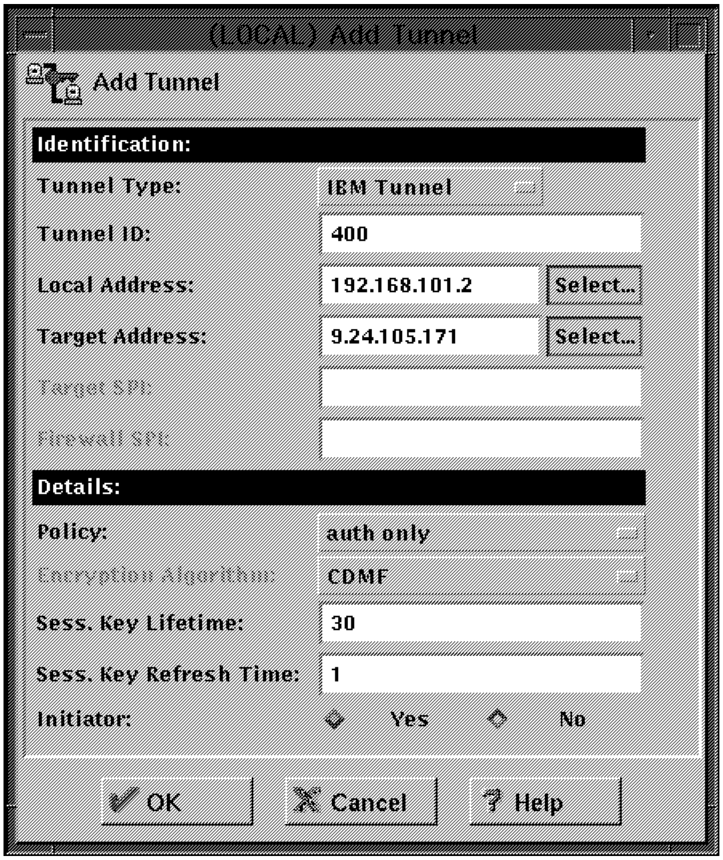
IP Address: 192.168.100.15

Subnet Mask: 255.255.255.255

Figure 69. Network Object for the Windows 95 System

The two objects just created replace the ones we created in the Branch Office scenario (see 5.3.3.2, “Creating the Network Objects in the First Firewall” on page 90 for FW1 and 5.3.3.7, “Creating the Network Objects in the Second Firewall” on page 96 for FW2).

The last modification in comparison to the Branch Office scenario has to be made in the tunnel definitions on FW1 (see 5.3.3.3, “Defining the Tunnel” on page 91). On FW2 the correct definition will be automatically created when you import the files from FW1. Figure 70 shows the panel for the tunnel definition on FW1.



(LOCAL) Add Tunnel

Add Tunnel

Identification:

Tunnel Type: IBM Tunnel

Tunnel ID: 400

Local Address: 192.168.101.2

Target Address: 9.24.105.171

Target SPI:

Firewall SPI:

Details:

Policy: auth only

Encryption Algorithm: CDMF

Sess. Key Lifetime: 30

Sess. Key Refresh Time: 1

Initiator: ● Yes ○ No

Figure 70. Tunnel Definition

All other steps for the firewall setup are identical to the Branch Office scenario setup. The above actions have the following effects:

1. Tunnel traffic is restricted to the AIX V4.3 and Windows 95 system.
2. Data is only authenticated and not encrypted because the encryption is covered by the nested tunnel between the AIX and the Windows 95 system.

Before we define the nested tunnel it is highly recommended to see whether the tunnel between the firewalls works. Issue an `admin_test` command on one firewall to see whether the tunnels can communicate. Make sure that your tunnel traffic is logged on the firewall. Are you able to ping the AIX V4.3 server from the Windows 95 system and are there corresponding tunnel packet entries in the firewall logs?

If the answer is yes, then your firewall tunnel is definitely working and you can continue by building the nested tunnel to ensure that there is an end-to-end encryption of the packets.

If the answer is no, you can find helpful information in Chapter 8, “Troubleshooting Your VPN” on page 151.

6.4 Configuration of the Nested Tunnel between the Tunnel Endpoints

In the following sections we build the manual tunnel between the tunnel endpoints. We start the setup on the AIX V4.3 server and proceed on the Windows 95 client.

6.4.1 Configuration of the AIX V4.3 Server

AIX V4.3 implements all current IPSec standards. Therefore you should be able to build a tunnel to any product that claims to fulfill the IPSec RFCs or Internet Drafts.

Our tests were mainly conducted on AIX V4.3.0, but we do recommend the use of AIX V4.3.1, because it includes additional IPSec functionality (see 4.2.2.6, “AIX V4.3.1 - New Functions” on page 69 for details).

Good to know

In order to be able to change a tunnel policy from SMIT in AIX 4.3.0, you have to apply a fix for bos.net.ipsec.rte (fix number U455606) which brings that module to level 4.3.1.0. What is not mentioned in the readme file of U455606 is that it also requires a fix for bos.net.tcp.client for level 4.3.1.0 (fix number U455614), which in turn requires a whole bunch of other fixes, but those are listed in the readme file of U455614. If you only apply U455606 but not U455614, the imptun command will fail. We recommend upgrading to AIX 4.3.1 all together where these problems have been resolved.

6.4.1.1 Prerequisite Steps

Before you start implementing a tunnel on AIX V4.3 please assure that the following prerequisites are met:

1. Basic AIX installation and TCP/IP setup has been completed.
2. Usage of coherent IP addresses in both secure networks. For example, you cannot use the same private IP addresses on both intranets. See 5.1.3, “Addressing Issues” on page 82 for details.
3. IP routing entries that enable traffic to the remote secure network. Normally the default route will point to the secure interface of the firewall. In general routing has been discussed in 5.1.4, “Routing Issues” on page 83.
4. Domain name serving will not be an issue as there is just one host in the remote secure network. Therefore an entry in the /etc/hosts file will probably be the easiest solution.
5. Installation of the IPSec filesets:
   - bos.net.ipsec.rte
   - bos.msg.LANG.net.ipsec where LANG is the desired language, such as en_US
   - bos.crypto for CDMF support (available on the World Trade Bonus Pak CD-ROM)

- bos.crypto_us for DES support (available on the U.S. Bonus Pak CD-ROM)
- bos.crypto-priv for 3DES support (available on the U.S. Bonus Pak CD-ROM of AIX V4.3.1)

Note: The Bonus Pak CD-ROM for the United States includes all transforms (DES and CDMF). Because of U.S. export regulations only CDMF is allowed to be shipped outside the U.S.

6.4.1.2 AIX V4.3 IP Security Setup

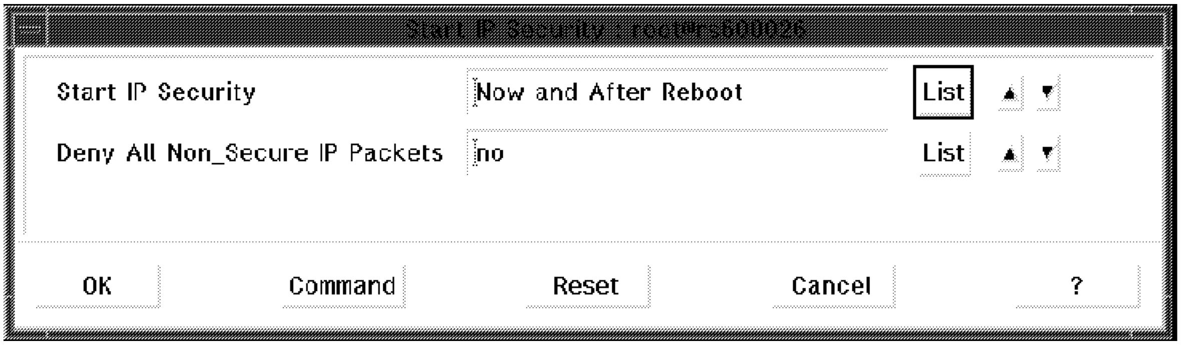
The following procedure sets up a tunnel on an AIX V4.3 server (rs600026).

Loading IP Security: After the installation of the IPSec filesets IP Security has to be loaded. This can be done separately for IPv4 and IPv6.

Note: We have only used IPv4, but the steps for IPv6 are exactly the same, they only differ in the SMIT path to follow:

- smit ipsec4 gets you to the IPV4 panels.
- smit ipsec6 allows for configuration of IP Security for IPv6.

Enter the SMIT fastpath `ips4_start` to load IP Security. Leave the defaults (see Figure 71) and press Enter. This will also enable the packet filtering function. It is important not to change the defaults. Otherwise you will deny traffic on your network interface.



Start IP Security : root@rs600026

Start IP Security: Now and After Reboot

Deny All Non_Secure IP Packets: no

Figure 71. Loading IP Security

Packet Filtering: Filtering can also be used without tunnels, to deny or permit any traffic on specific criteria such as source and destination IP addresses and masks. Similar to the eNetwork Firewall for AIX other possible options include:

- Protocol
- Port number
- Direction
- Fragmentation control
- Routing
- Tunnel
- Interface

In addition you can specify on a per rule basis whether logging should be done or not.

If you define a tunnel, AIX will automatically configure the filter rules needed to allow any traffic through the tunnel. Of course the filters can be changed afterwards if there are specific needs to be met. Go to the SMIT IPSec main panel (smit ipsec4), then select **Advanced IP Security Configuration** and

Configure IP Security Filter Rules. Modifications done within this series of menus have to be activated by choosing **Activate/Update/Deactivate IP Security Filter Rule**.

You can check the contents of your rule base via the SMIT path **List Active IP Security Filter Rules** or the lsfilt command. You can find some sample filter rules in the file /usr/samples/ipsec/filter.sample. For our scenario we do not need to touch the filters at all, because the auto-generated filter rules allow any traffic through the tunnel already.

Logging Facilities: This section describes the configuration and format of system logs relating to IPSec. As hosts communicate with each other, the transferred packets may be logged to the system log daemon, syslogd. Other important messages about IPSec will appear as well. An administrator may choose to monitor this logging information for traffic analysis and debugging assistance. Follow these steps for setting up the logging facility. The first step is to modify the /etc/syslog.conf file and add the following entry:

```
local4.debug /var/adm/ipsec.log
```

Of course you can choose any other location on your disk. We recommend to create a new file system (for instance /var/log) and put the file in there.

Note: The logging of filter events can create significant activity at the IPSec host and can consume large amounts of storage.

Use the local4 facility to record traffic and IPSec events. Standard AIX priority levels apply. We recommend setting the priority level of debug until traffic through IPSec tunnels and filters show stability and proper movement.

Once you have added the entry, save /etc/syslog.conf, go to the directory you specified for the log file and create an empty file with the same name. In the case above, you would change to the /var/adm directory and issue the command: `touch ipsec.log`. After these two steps have been completed, issue a refresh command to the syslogd subsystem: `refresh -s syslogd`.

While creating filter rules for your host, if you would like packets matching a specific rule to be logged, set the logging option for the rule to yes. Finally, turn on packet logging and start the IPSec log daemon using the following command: `mkfilt -g start`. You may stop packet logging by issuing the following command: `mkfilt -g stop`. The corresponding SMIT panels can be found on the advanced IPSec menus under **Start/Stop IP Security Filter Rule Log**.

Please see 8.3.3, “Logging” on page 161 for a detailed description of the AIX V4.3 IPSec log entries.

Tunnel Definition: The next step is to define a tunnel with the following main features:

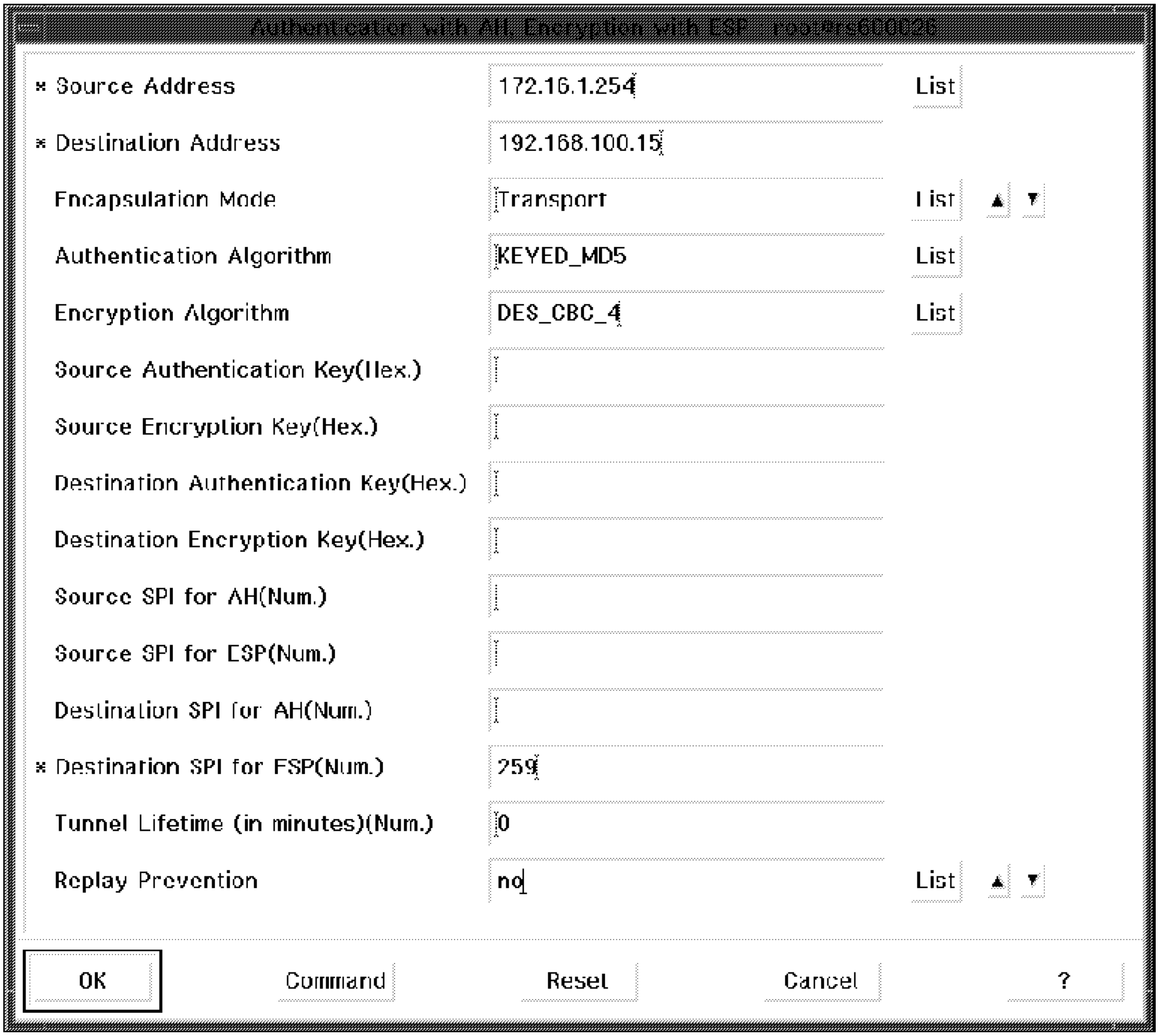
1. Tunnel type: Manual tunnel
2. SA type: Transport mode
3. Policy: Encryption only

From the SMIT main IPSec panel choose **Basic IP Security Configuration**, **Add IP Security Tunnel**, **Use Manual Session Key Refresh Method (Manual Tunnel)**, **Host-Host (Manual Tunnel)** and then **Authentication with AH, Encryption with ESP**. See Figure 72 on page 124 for the contents of the definition panel.

There is no possibility to choose Encryption Only with the old header format if you use SMIT. (It is possible if you use the command line, but in SMIT this policy will use the new headers.) Therefore, we have to create a tunnel with the Encrypt and then Authenticate policy and change it to Encryption Only afterwards. We chose **DES CBC 4** just because we wanted to test all available transforms. In a real environment you will always choose the strongest available transform (which would be DES CBC 8 if your partner is the Windows 95 eNetwork Communications Suite). You cannot enter a value for the tunnel ID; it will be automatically assigned by the system.

Replay prevention is not possible if your partner uses the original RFC headers as in our case. It is only possible if your partner uses the new Internet Draft header formats. In our case the AH transform does not matter, because we will change the policy to Encryption Only.

A tunnel lifetime of zero (unlimited) is good from an administration point of view, because there is no key distribution problem. Unfortunately it is no valid solution from our point of view, because there is a security problem if you do not refresh/change the keys from time to time. Until there is a final solution (see Chapter 10, “The Internet Key Exchange (IKE) Protocol” on page 193) you have to live with this administration overhead. We strongly recommend you change your keys on a regular basis. See 6.6, “Manual Key Distribution” on page 131 for some thoughts on manual key distribution.



Authentication with AH, Encryption with ESP - root@rs600026

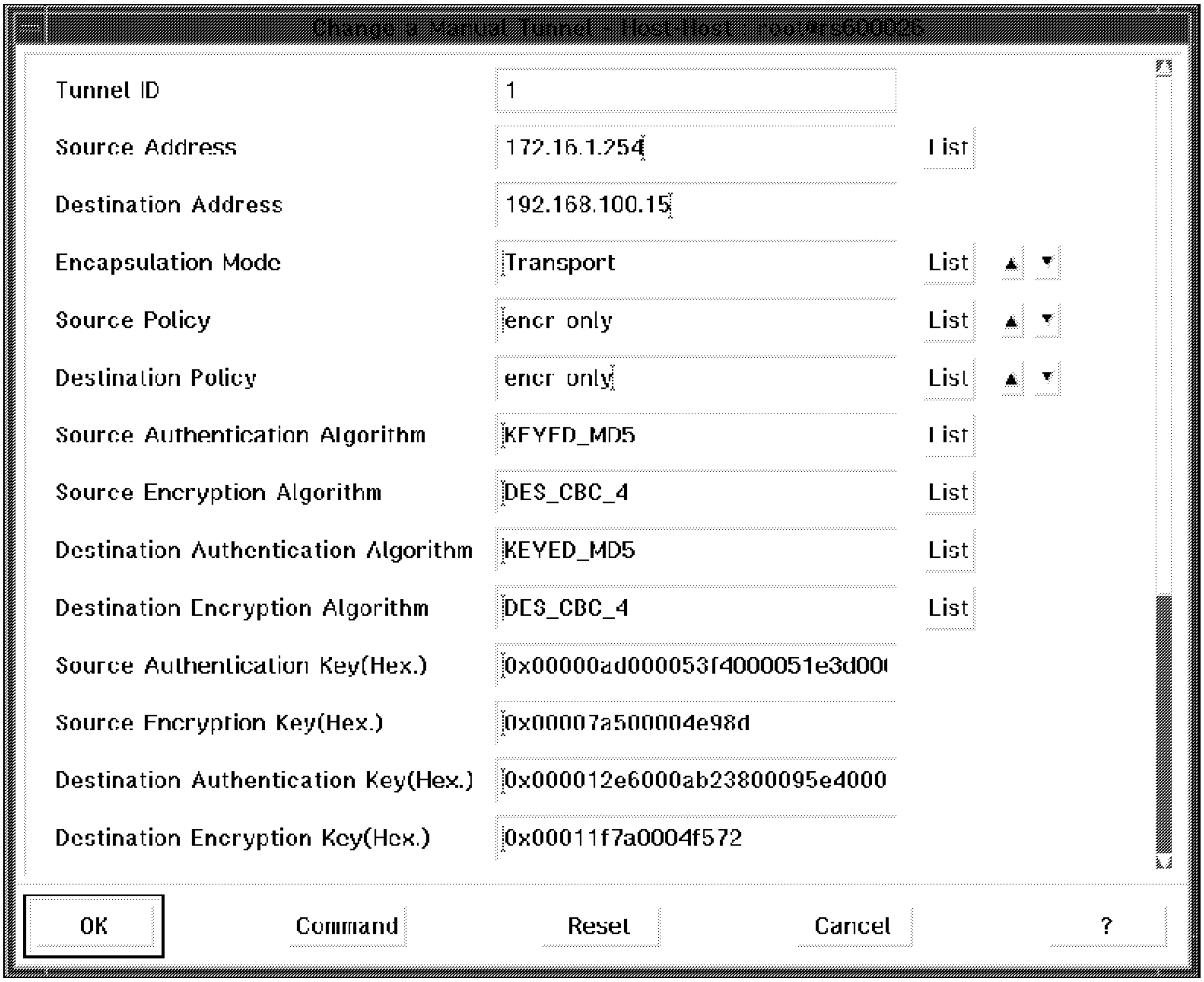
* Source Address	172.16.1.254
* Destination Address	192.168.100.15
Encapsulation Mode	Transport
Authentication Algorithm	KEYED_MD5
Encryption Algorithm	DES_CBC_4
Source Authentication Key(Hex.)	
Source Encryption Key(Hex.)	
Destination Authentication Key(Hex.)	
Destination Encryption Key(Hex.)	
Source SPI for AH(Num.)	
Source SPI for ESP(Num.)	
Destination SPI for AH(Num.)	
* Destination SPI for ESP(Num.)	259
Tunnel Lifetime (in minutes)(Num.)	0
Replay Prevention	no

Figure 72. Tunnel Definition

In order to make life easier we only specified necessary parameters. Therefore we need to find out the key and SPI values so that we will be able to match them in the partner tunnel definition. Issuing a: `lstun -p manual -a` command will list the values of all manual tunnels defined on the AIX system. The -a option is required for an inclusion of the values for the keys.

Another possibility is to use the SMIT change tunnel panel. The keys are listed there as well.

Now we need to change the policy of this tunnel. On the Basic IP Security Configuration panel choose **Change IP Security Tunnel**, **Use Manual Session Key Refresh Method (Manual Tunnel)** and **Host-Host**. Then you need to select the tunnel that you have just created before. Change the source and the destination policy to encr only (see Figure 73 on page 125).



Change a Manual Tunnel - Host-Host   root@rs600026

Tunnel ID	1
Source Address	172.16.1.254
Destination Address	192.168.100.15
Encapsulation Mode	Transport
Source Policy	encr only
Destination Policy	encr only
Source Authentication Algorithm	KEYED_MD5
Source Encryption Algorithm	DES_CBC_4
Destination Authentication Algorithm	KEYED_MD5
Destination Encryption Algorithm	DES_CBC_4
Source Authentication Key(Hex.)	0x00000ad000053f4000051e3d00
Source Encryption Key(Hex.)	0x00007a500004e98d
Destination Authentication Key(Hex.)	0x000012e6000ab23800095e4000
Destination Encryption Key(Hex.)	0x00011f7a0004f572

Figure 73. Changing the Tunnel Definition

The necessary packet filters were created automatically when you defined the tunnel. You can list the filters for instance by issuing a lsfilt command. The lines below contain only the filter rules that are necessary for our tunnel definition.

```
Rule 2:
Rule action         : permit
Source Address      : 0.0.0.0
Source Mask         : 0.0.0.0
Destination Address : 0.0.0.0
Destination Mask    : 0.0.0.0
Source Routing      : yes
Protocol            : ah
Source Port         : any 0
Destination Port    : any 0
Scope               : both
Direction           : both
Logging control     : no
Fragment control    : all packets
Tunnel ID number    : 0
Interface           : all
Auto-Generated      : yes

Rule 3:
Rule action         : permit
Source Address      : 0.0.0.0
Source Mask         : 0.0.0.0
Destination Address : 0.0.0.0
Destination Mask    : 0.0.0.0
Source Routing      : yes
Protocol            : esp
Source Port         : any 0
Destination Port    : any 0
Scope               : both
Direction           : both
Logging control     : no
Fragment control    : all packets
Tunnel ID number    : 0
Interface           : all
Auto-Generated      : yes

Rule 4:
Rule action         : permit
Source Address      : 172.16.1.254
Source Mask         : 255.255.255.255
Destination Address : 192.168.100.15
Destination Mask    : 255.255.255.255
Source Routing      : yes
Protocol            : all
Source Port         : any 0
Destination Port    : any 0
Scope               : both
Direction           : outbound
Logging control     : no
Fragment control    : all packets
Tunnel ID number    : 1
Interface           : all
Auto-Generated      : yes

Rule 5:
Rule action         : permit
Source Address      : 192.168.100.15
Source Mask         : 255.255.255.255
Destination Address : 172.16.1.254
Destination Mask    : 255.255.255.255
Source Routing      : yes
Protocol            : all
Source Port         : any 0
Destination Port    : any 0
Scope               : both
Direction           : inbound
Logging control     : no
Fragment control    : all packets
Tunnel ID number    : 1
Interface           : all
Auto-Generated      : yes
```

The auto-generated filters will be activated automatically.

Tunnel Activation: We finish the configuration on the AIX V4.3 server by activating the tunnel definition. Go to the SMIT panel Basic IP Security Configuration and select **Activate IP Security Tunnel**, then select the tunnel you have just defined from the list and press **OK**.

6.4.2 Configuration of the eNetwork Communications Suite Client

The Windows 95 eNetwork Communications Suite client implements the original IPSec RFCs. It does not support the new Internet Draft headers. During our tests we used Version 1.1 of the product. We used the U.S. version, because encryption is not available in the international version.

6.4.2.1 Prerequisite Steps

Before you start implementing the tunnel on Windows 95 please assure that the following prerequisites are met:

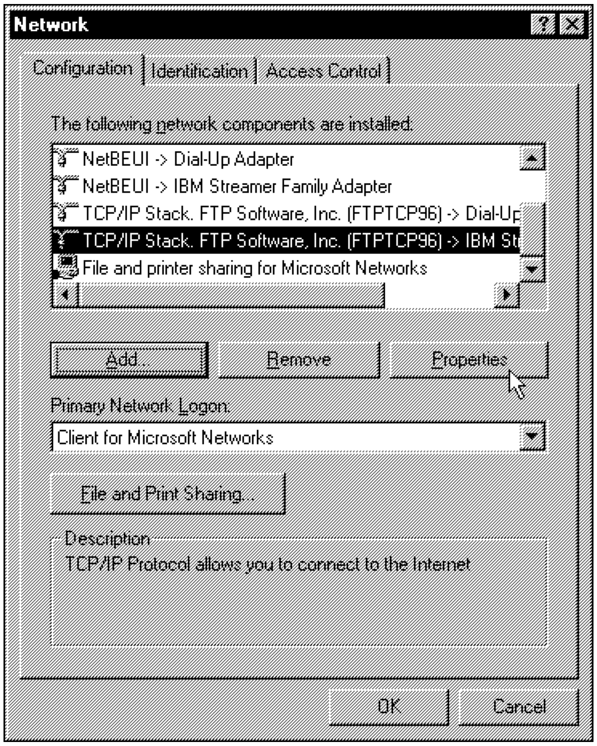
1. Basic Windows 95 installation and setup has been completed.
2. Basic eNetwork Communications Suite installation and TCP/IP setup has been completed.
3. Usage of coherent IP addresses in both secure networks. For example, you cannot use the same private IP addresses on both intranets. See 5.1.3, “Addressing Issues” on page 82 for details.
4. IP routing entries that enable traffic to the remote secure network. Normally the default route will point to the secure interface of the firewall. In general routing has been discussed in 5.1.4, “Routing Issues” on page 83.
5. Domain name serving will not be an issue as there is just one host in the remote secure network. Therefore an entry in the windowsftptcp96hosts file will probably be the easiest solution.

Note: For detailed information on the eNetwork Communications Suite software we recommend the redbook *Exploring the IBM eNetwork Communications Suite*, SG24-2111.

6.4.2.2 eNetwork Communications Suite IP Security Setup

The following procedure sets up the end-to-end tunnel to the AIX V4.3 server (rs600026):

1. Click on the **Start** button, then select **Settings** and **Control Panel**. Once the Control Panel folder has opened, double-click on the **Network** object. From the Configuration tab, select the component called **TCP/IP Stack. FTP Software...** and then select **Properties** (see Figure 74 on page 128).



The following network components are installed:

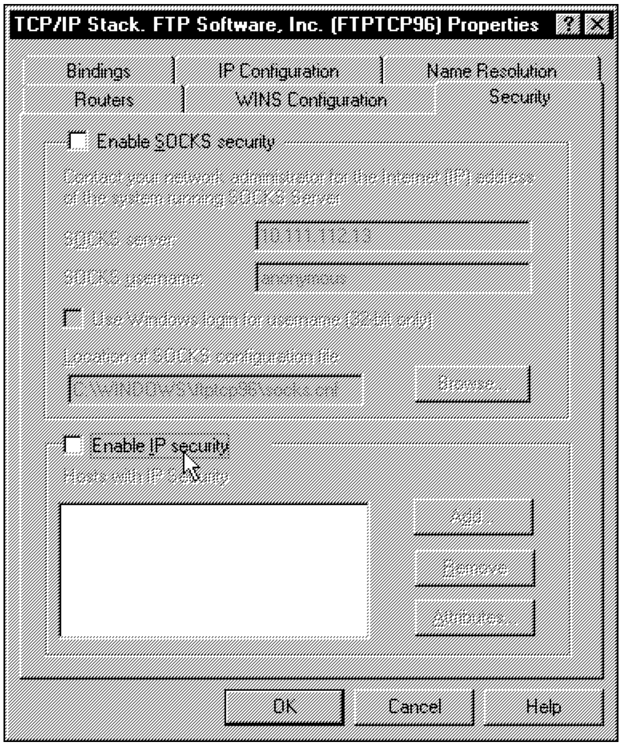
- NetBEUI -> Dial-Up Adapter
- NetBEUI -> IBM Streamer Family Adapter
- TCP/IP Stack. FTP Software, Inc. (FTPTCP96) -> Dial-Up
- TCP/IP Stack. FTP Software, Inc. (FTPTCP96) -> IBM St
- File and printer sharing for Microsoft Networks

Primary Network Logon: Client for Microsoft Networks

Description
TCP/IP Protocol allows you to connect to the Internet

Figure 74. Network Panel

2. From the configuration notebook, select the tab marked **Security**. Check the box marked **Enable IP Security** (see Figure 75).



□ Enable SOCKS security

Contact your network administrator for the Internet (IP) address of the system running SOCKS Server.

SOCKS server: 10.111.112.13

SOCKS username: anonymous

□ Use Windows login for username (32-bit only)

Location of SOCKS configuration file: C:\WINDOWS\ftptcp96\socks.cnf

□ Enable IP security

Hosts with IP Security

Figure 75. IP Security Panel

3. You will be presented with a dialog box asking for an IP security number. Type any combination of numbers and letters, until the **OK** push button gets

enabled. Then click on this button. This information is a seed for generating the security association numbers. You will be presented with this panel each time you reboot your system to re-initialize the random number seed. This procedure remains active as long as you have IP Security enabled.

4. Click on the **Add** button to start the tunnel definition. Now you need to enter values that match the security association already defined on the AIX V4.3 system (see Figure 76 on page 130 and the output from the lstun command described there).

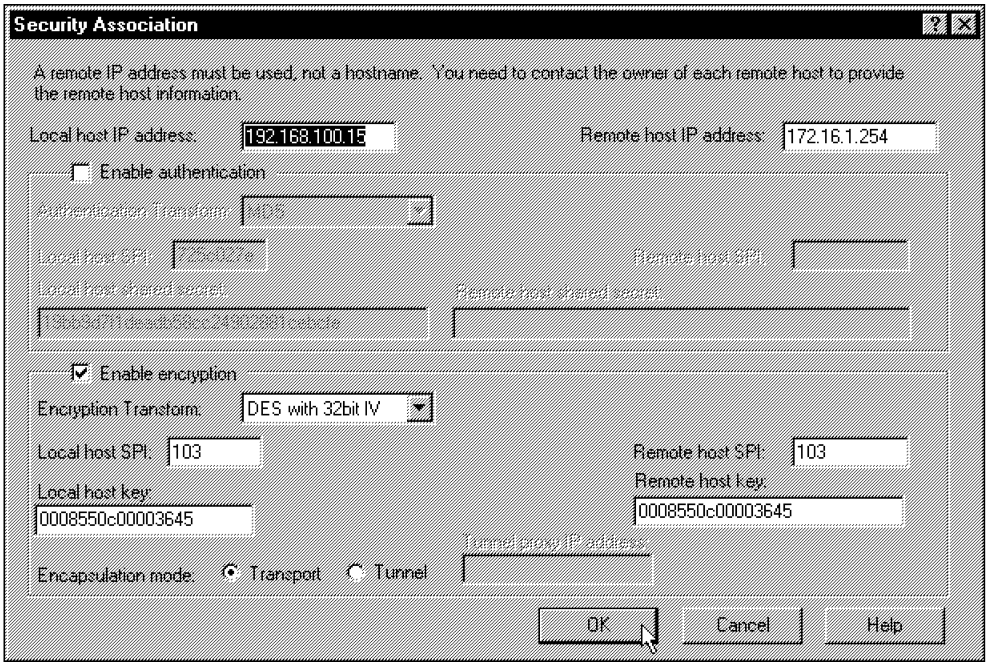
   Notes:

   a. eNetwork Communications Suite expects hexadecimal values for the SPI fields. Therefore you have to convert the decimal AIX V4.3 value to the corresponding hexadecimal value before entering it on the panel below. In our case the decimal AIX V4.3 SPI value 259 corresponds to hexadecimal 103 on the eNetwork Communications Suite client.

   b. To make it easier to identify matching configuration values between the two systems, take a look at Table 4. Note that on the AIX side the terminology between field names in the SMIT panels (Add IP Security Tunnel and Change IP Security Tunnel) and the output from the: `lstun -a` command are different in some cases.

Table 4. Matching Configuration Items between AIX V4.3 and eNetwork Communications Suite

AIX V4.3	**eNetwork Communications Suite**
source IP	remote IP
destination IP	local IP
source SPI	remote SPI
destination SPI	local SPI
source (SMIT) = send (lstun) key	local key
destination (SMIT) = receive (lstun) key	remote key
source (SMIT) = send (lstun) transform	local transform
destination (SMIT) = receive (lstun) transform	remote transform



A remote IP address must be used, not a hostname. You need to contact the owner of each remote host to provide the remote host information.

Local host IP address: 192.168.100.15 Remote host IP address: 172.16.1.254

□ Enable authentication

Authentication Transform: MD5

Local host SPI: 72bc027e Remote host SPI:

Local host shared secret: 13bb9d71deadb58cc2490281cebcfe

Remote host shared secret:

☑ Enable encryption

Encryption Transform: DES with 32bit IV

Local host SPI: 103 Remote host SPI: 103

Local host key: 0008550c00003645

Remote host key: 0008550c00003645

Encapsulation mode: ● Transport ○ Tunnel

Tunnel proxy IP address:

Figure 76. Tunnel Definition

5. Click on **OK** to save this security association. You have to specify a password to be used for additional authorization as long as IP Security is enabled. Therefore from now on there will be two additional panels at reboot: one for the random combination of letters and numbers to initialize the random number seed and one for the IPSec logon password.

6. You need to reboot after every change of the IPSec configuration.

   Note: All defined tunnels will be automatically activated at boot time. There is no possibility to activate or deactivate tunnels manually.

6.5 Testing the Tunnel

Issue a ping from the Windows 95 client to the AIX V4.3 server. If you get a response the nested tunnel works fine. You can prove this on both sides.

On the AIX V4.3 server the easiest possibility is to enable logging for all tunnel related filter rules. You should then see the tunnel traffic in your log files. A more complicated alternative would be IP tracing.

There are no IPSec logging facilities or filter rules on the eNetwork Communications Suite client. You can use the IPTrace application of the eNetwork Communications Suite protocol stack to verify that IPSec is being used. Actually it is a very user-friendly implementation. For more information on the eNetwork Communications Suite see the redbook *Exploring the IBM eNetwork Communications Suite*, SG24-2111.

You can also check on the firewalls. If logging is enabled for all tunnel-related filter rules, you will see different protocols for different parts of the end-to-end tunnel:

1. ESP packets for the intranet parts of the tunnel

2. AH packets for the part between the firewalls (where the ESP packets get a new IP header because the IBM tunnel between the firewalls uses tunnel mode)

Given the information above, we can now draw the outline of a packet in this nested tunnel:

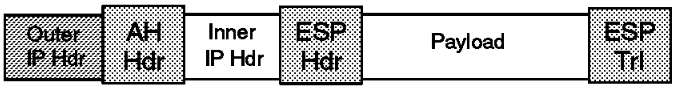


Figure 77. ESP Packet Nested Inside AH Tunnel

6.6 Manual Key Distribution

As we mentioned in Chapter 4, "IBM eNetwork VPN Solutions" on page 59 the current IPSec RFCs provide no key management functionality. Therefore, unfortunately, the only short term solution for manual tunnels is manual key distribution. In our opinion this currently limits manual tunnels to small scale implementations. This will change later this year with the introduction of Internet Key Exchange (IKE) (see Chapter 10, "The Internet Key Exchange (IKE) Protocol" on page 193).

6.6.1 Using Mail

Most methods of exchanging keys are non-secure. Cutting and pasting the key into an e-mail message would be the simplest and most efficient manual method of key distribution, but unless the mail is secure itself, it is open to the possibility of being intercepted. Many e-mail packages keep the data in plain text on the host system, or server. This makes the secret key information vulnerable.

Of course sending conventional mail (for instance a diskette containing the keys) is also a valid way but it usually introduces a time problem. Depending on the value for the session key lifetime it could well be that the keys expire before the diskette arrives at your partner′s location. You could send many keys at once and later on tell your partner on the phone to change to the second, third or whatever key.

6.6.2 Using Secure Socket Layer (SSL)

A possible means of exchanging keys in a secure manner is the use of a separate encrypted communications channel. An example of an encrypted communications channel is Secure Sockets Layer (SSL). SSL is a client/server based method of secure communication. SSL was developed by Netscape Communications Corporation as an open, non-proprietary protocol.

The SSL client could be for instance Netscape Navigator. For the server, IBM offers the Internet Connection Secure Server (ICSS). ICSS is available under a separate license agreement. For more information on SSL, refer to the following URL: http://home.netscape.com/assist/security/ssl/index.html.

6.6.3 Using Pretty Good Privacy (PGP)

Another way to send keys to your partner is by employing *Pretty Good Privacy (PGP)*. PGP is a freely available program that uses public-key cryptography to create encrypted and authenticated messages.

You can download the international version of PGP from `http://www.pgpi.com`. If you are based in the U.S., you should use `http://www.pgp.com`.

For a detailed description on how to use PGP on an RS/6000 see section 7.6 of the redbook *Protect and Survive Using IBM Firewall 3.1 for AIX*, SG24-2577.

6.7 Variations of the Business Partner/Supplier Network Scenario

There are two variations to the business partner/supplier scenario that we could think of but did not elaborate on:

1. The first one is actually the ″normal″ solution: use of an ESP new header format protocol, which allows authentication within the ESP protocol.
2. The second variation: if both sides use official Internet addresses and do not need to hide their intranet structure, there is no need for a FW-to-FW tunnel. (One would just need the proper filter rules on the firewall.)

Chapter 7. Remote Access Scenario

With the advent of tele-working, remote access to corporate networks is increasingly important these days. The traditional way of deploying modem pools and remote access servers is expensive because of the dedicated equipment needed and especially because of the long-distance telephone costs involved. As the Internet has become virtually omnipresent, just a local phone call away, the remote access costs can be greatly reduced by using the Internet as the access infrastructure to the corporate network.

In this chapter we examine how remote users can exploit IPSec features to establish secured connections to their corporate intranets over the Internet.

For this scenario, let′s assume that company A has procured Internet access from an ISP and wants to enable its mobile work force to access the resources located in the corporate network over the Internet. For simplicity we do not consider other possible connections, such as Internet access or other VPN scenarios. These issues can be dealt with separately. The techniques described here can also be applied to secure traditional dial-in connections.

7.1 Design Considerations

The major issue to be addressed is the inherently dynamic nature of this scenario. Typically SAs cannot be pre-configured because the clients′ addresses cannot be predicted. ISPs assign addresses dynamically. At some ISPs it is possible to request fixed addresses for dial-in connections, but only at an extra charge.

We need to be able to identify the remote client by its name rather than by its IP address. ISAKMP/Oakley has this ability, but it is not yet implemented in IBM products. The only way to accomplish identification by name today is to use the dynamic tunnel feature of the IBM Firewall for AIX at the corporate gateway. This choice gives us two possibilities for the client operating system: Windows 95 with the IPSec Client and OS/2 with TCP/IP V4.1.

Note: IBM has plans to offer an IPSec client on Windows NT later this year.

To facilitate effective tunnel management, with dynamic tunnels most of the tunnel parameters are auto-generated, much like in the case of IBM tunnels. The filter rules associated with a dynamic tunnel are automatically effective at client connection time and cease operation when the client closes the tunnel. These dynamic filter rules are always checked *before* the static filter rules and cannot be modified. Advantages of this mode of operation are the ease of use and the guaranteed functionality. However, the dynamic filter rules permit all traffic from the client to the whole intranet including the firewall itself and there is no way to change that. The reason behind this is that a remote client normally needs the same access to the resources as if it were actually in the intranet. The security policy for the secure network, not the firewall, is what determines the access for the remote client once connected and authenticated.

If the clients′ addresses are known (preassigned by the ISP), then we are not limited to dynamic tunnels and the clients mentioned. Basically any kind of valid combination listed in 4.4, “Interoperability between the IBM Solutions” on page 75 will work.

Another important design point is whether to extend IPSec tunnels to the hosts in the intranet or not. Most companies trust their intranets; for them there is no reason to do so. This approach has the advantage of a much smaller number of SAs to be managed. Also, the end systems do not need to be modified to support IPSec.

Sometimes very stringent security regulations are in place and the intranet is untrusted. In this case the IPSec tunnels should go directly to the destination host. IPSec should be deployed to those hosts and a large number of SAs should be managed. These factors could have significant cost and system management implications. A setup like this which is also shown in Figure 31 on page 54 cannot (yet) be facilitated using the current IBM IPSec clients.

Note: We tried to implement the nested tunnels shown in Figure 31 on page 54with the only plausible client, the eNetwork Communications Suite TCP/IP stack for Windows 95. We did not succeed and the trace analysis showed the cause; this client does not support AH in tunnel mode. The implementation of ISAKMP/Oakley will solve this problem.

Extending a tunnel from the client to the server also could make sense in another, very special situation: when the client is a *foreign* one, for example a traveling business partner′s notebook that is allowed to connect to a corporate server. This setup resembles to the business partner/supplier scenario, the difference being that all tunnels originate from the client itself.

The table below consolidates the capabilities of the current IBM IPSec implementations in the remote access scenario.

Table 5. IPSec Product Choices for Remote Access

Client	**To Gateway**	**To Host**
Dynamic IP address at the client	IBM Firewall for AIX + Windows 95 IPSec Client or OS/2 TCP/IP 4.1	None
Fixed IP address at the client	Any valid combination	None

7.1.1 Data Confidentiality and Authentication

It is obvious that company A wants the dial-in traffic to be encrypted. Authentication is also needed because the corporate firewall must admit only traffic from the remote clients. Thus, either ESP tunnels with authentication option or combined AH-ESP tunnels should be used. The latter is the only possibility to provide both authentication and encryption with the current IBM Firewall.

7.1.2 Addressing and Routing Issues

Unlike in the Branch Office connection or business partner/supplier scenarios, here we have one endpoint of the tunnels in the Internet. The clients will have automatically assigned public IP addresses by the ISP at connect time. These are routable everywhere. The router installed by the ISP at company A′s site knows how to route to the Internet. Therefore, the only requirement for the internal routers is to have routes that direct Internet traffic to the corporate firewall, which in turn routes to the ISP′s router. This should be the case anyway.

The IPSec code at the dial-in clients should be capable of differentiating between the corporate traffic which is to be tunneled and the ordinary Internet traffic that requires no special treatment. If they sent all traffic through the tunnel, then the remote user would loose the ability to access Internet resources while operating that tunnel, because the firewall normally would drop the packets retrieved from a tunnel that have non-secure source and destination addresses.

The addressing scheme of the intranet needs no modification to support dial-in clients. If the intranet uses private addresses, it will still be reachable, because packets with private IP addresses are tunneled and the tunnel endpoints have public addresses. Only the subnets with direct connection to the Internet need to have public addresses. This is no new requirement.

7.1.3 Multiprotocol Support

If protocols other than IP should be supported for the remote clients, then besides IPSec an appropriate tunneling protocol that will carry the non-IP payload must be supported by the firewall and by the ISP′s Point of Presence. However, these protocols do not offer robust cryptographic features comparable to IPSec. Therefore the solution is to use IPSec to protect the traffic that flows in the multiprotocol tunnel.

In this case a viable choice is L2TP which is likely to be supported by more and more ISPs. Note that the non-IP protocol must not only be supported at the remote client and at the destination server, but also at the firewall. Otherwise the firewall would have no means to send the decapsulated non-IP payload to its ultimate destination.

With L2TP, the PPP connection that was in place between the remote client and the ISP is now extended to the corporate firewall. This results in the client and the firewall being on the same IP subnet and allows for the firewall to assign the client′s address itself.

7.1.4 Summary: Remote Access

This application of VPN technology replaces existing direct dial-in lines to the corporate network and instead uses the Internet as the access infrastructure. Here are the major design considerations:

- The solution does not require changes at the servers in the corporate network unless the dial-in traffic is to be protected against attacks on the intranet as well. However, clients have to support the IPSec protocols.
- Because client addresses are typically dynamic, the IBM Firewall′s dynamic tunnel is the adequate tunnel type. This does not allow for tunnels to extend back to the servers in the corporate intranet, but in practice this is not a great disadvantage. ISAKMP/Oakley will remove this limitation. Manual tunnels are feasible only with fixed remote client IP addresses.
- The dial-in traffic will be encrypted and authenticated. Any traffic that cannot be authenticated will be rejected by the firewall.
- There are no special routing or addressing issues. Existing intranet addresses can be used ″as is″.
- Existing packet filtering rules, if any, do not interfere with the dynamic filter rules. They can be used without modification.

- Explicit filter rules to protect the corporate intranet against non-VPN traffic are not required because the IPSec authentication will provide this protection.

Note: We would like to remark here that the term *corporate firewall* used in the previous paragraphs could be replaced with the term *router* if such a box offers equivalent functionality, such as dynamic tunnels or ISAKMP/Oakley.

7.2 Scenario Setup

Our test network is shown in Figure 78.

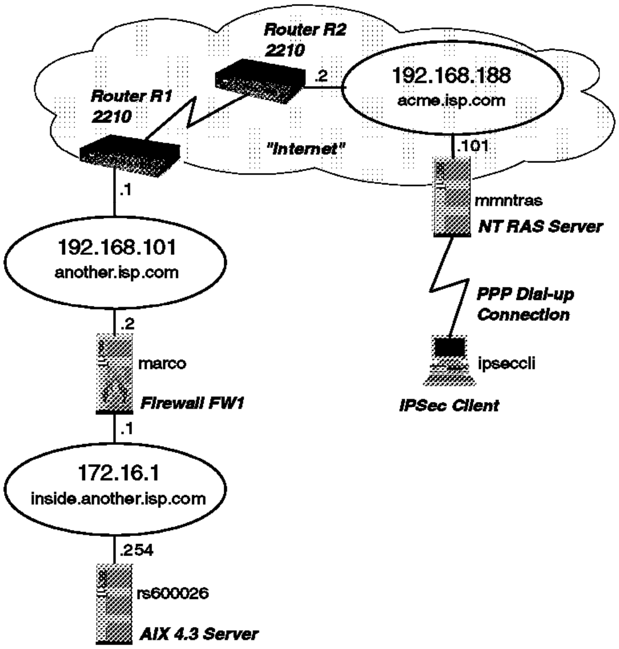


Figure 78. Remote Access Scenario

The Internet consists of the routers R1 and R2 interconnected with a serial link and the ISP′s Point of Presence, the NT Server mmntras. It is connected through the subnet acme.isp.com to router R2. The NT Server runs the Remote Access Service (RAS). RAS is enabled for IP only and assigns addresses to the dial-in clients dynamically using DHCP. The RAS server is configured to accept dial-in requests using PPP using any kind of authentication.

Note: The dial-in infrastructure is a given in this scenario, therefore we do not elaborate on it.

The dial-in client is the PC ipseccli running either Windows 95 or OS/2 with the appropriate IPSec client code. We describe in detail the setup of the OS/2 IPSec client, because the Windows 95 client has already been covered in the *Protect and Survive Using IBM Firewall 3.1 for AIX*, SG24-2577 redbook.

The corporate intranet is the subnet inside.another.isp.com, protected by the firewall FW1 (marco) and the DMZ another.isp.com. The firewall runs AIX 4.2.1 and IBM Firewall for AIX 3.1.1.5. The server rs600026 runs the business application that the remote client wishes to use. It is an AIX 4.3 machine. Its IPSec capabilities are used when the tunnel from the client extends to it.

We built a dynamic, encrypted and authenticated tunnel between the remote client and the firewall. All traffic between the corporate intranet and the client flows through this tunnel. In this case we used the IPSec client of OS/2 TCP/IP 4.1, which is specially designed to use dynamic tunnels. This is outlined in Figure 37 on page 87.

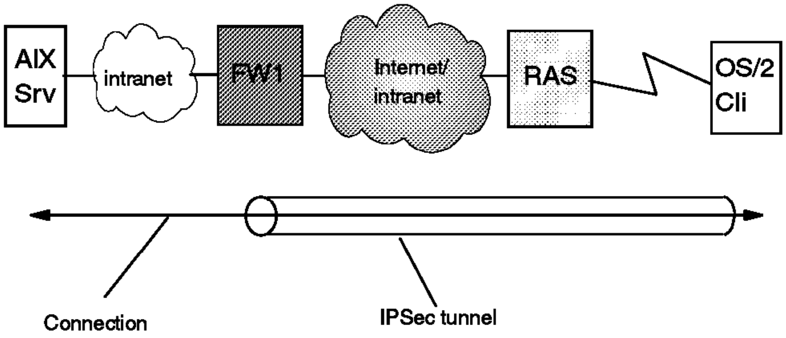


Figure 79. Remote Access Scenario - Secure Tunnel Between Remote Client And Corporate Firewall

7.3 Configuring the Components

The following sections guide you through the configuration of the remote access scenario. After listing the prerequisites we provide an overview of the configuration steps followed by the detailed procedure description. We also include information to help you understand the working of the tunnel. We assume that you are familiar with the information presented in Chapter 4, “IBM eNetwork VPN Solutions” on page 59.

7.3.1 Prerequisite Steps

The following prerequisites need to be met at the machines involved in this scenario:

1. Basic firewall installation and setup has completed, including network objects for your non-secure firewall interface and for your secure network.

Don′t Forget

You need to create an SSL key ring using the mkkf command, because the remote clients use an SSL connection to retrieve the dynamic tunnel policy from the firewall. Edit the /etc/security/rcsfile.cfg to point to your keyfile created with mkkf and make sure that the sslrctd daemon that handles the SSL requests is running. For complete instructions on how to use the mkkf command see Chapter 5, ″Using the Make Key File Utility (MKKF)″ in the *eNetwork Firewall for AIX, Reference Guide*, SC31-8418.

2. Firewall code has been updated to the latest level. We have worked with V3.1.1.5 and had no problems.

Good to know

Due to changes in the remote user logon validation between V3.1.1.5 and V3.2.1 of the eNetwork Firewall for AIX, the OS/2 TCP/IP V4.1 IPSec Client will fail to connect to the firewall running at V3.2.1 level. By the time this redbook was ready to print, IBM development was working on a fix for this problem.

The description of the OS/2 TCP/IP V4.1 IPSec Client in this chapter is accurate and functional when connecting to an eNetwork Firewall for AIX V3.1.1.5. We received a preversion of the aforementioned fix and could then also connect the OS/2 TCP/IP V4.1 IPSec Client to an eNetwork Firewall for AIX V3.2.1, but only with user ID root.

3. IP forwarding has been enabled at the firewall (enter `no -o ipforwarding=1`). Otherwise the packets cannot be routed between the secure and non-secure interfaces of the firewall.
4. The routing tables in the servers in the intranet are correct. They need to point to the firewall for addresses in the Internet.
5. The OS/2 TCP/IP V4.1 IPSec Client and the dialer have been installed at the client. These components are part of TCP/IP V4.1.

Notes:

1. If you need more information on how to set up the eNetwork Firewall for AIX we recommend the redbook *Protect and Survive Using IBM Firewall 3.1 for AIX*, SG24-2577.
2. For prerequisites and installation instructions for the OS/2 TCP/IP V4.1 IPSec Client see the documentation that comes with the product. Please see 4.3.3, “OS/2 TCP/IP V4.1 IPSec Client” on page 73 for more information on TCP/IP V4.1 for OS/2.
3. We have also successfully tested the remote access to an eNetwork Firewall for AIX V3.2.1 using the Windows 95 IPSec Client supplied with that level of the firewall, but we have not elaborated on it because it is described in the redbook *Protect and Survive Using IBM Firewall 3.1 for AIX*, SG24-2577, and also in the *eNetwork Firewall for AIX, User*, GC31-8419.

7.3.2 Setup Overview

In order to get a dynamic tunnel to work you need to accomplish these tasks:

- On the firewall:
  1. Create the objects for the dial-in client:
     a. A proxy user with non-secure IP authentication set to password.
     b. A network object referring to the previously created proxy user.
  2. Define the dynamic tunnel.
  3. Create the connection that allows SSL traffic from the world to the non-secure adapter.
- On the remote client:
  1. Configure the dial-in connection to the ISP.
  2. Define the IPSec connection to the firewall.

7.3.3 Setup Details

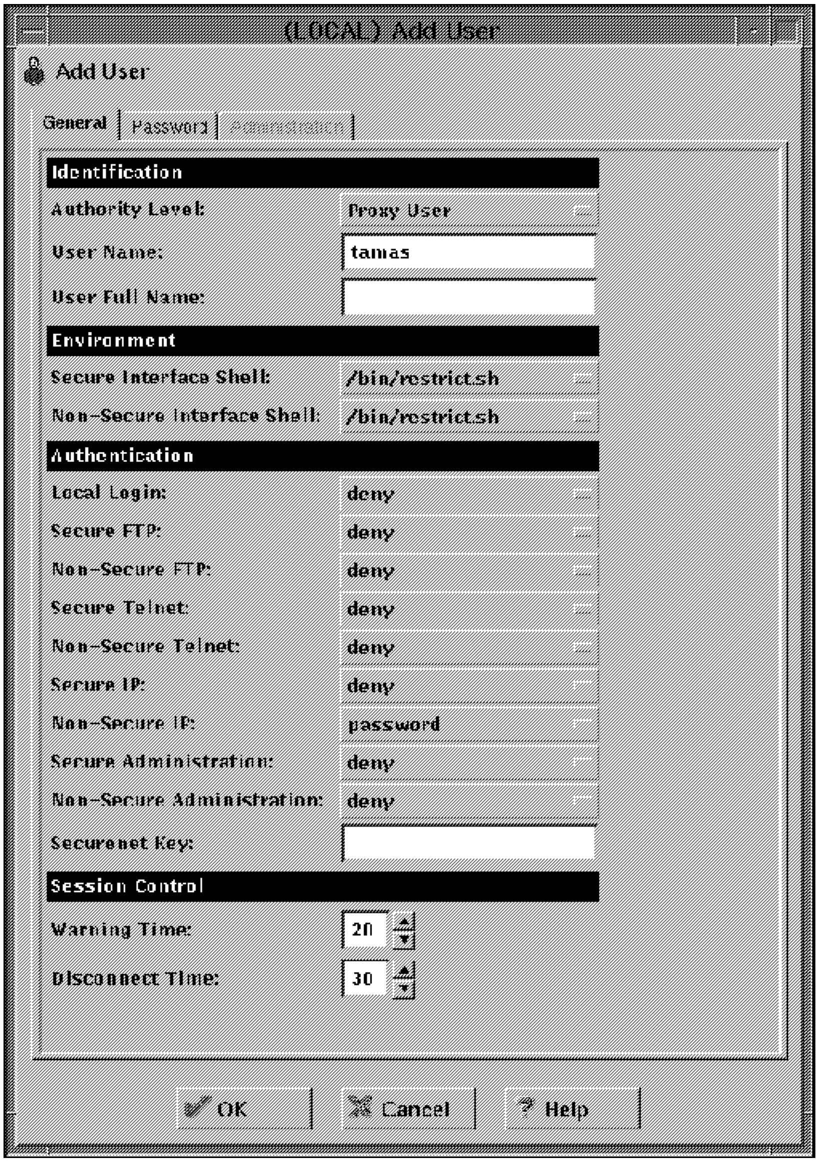
Let's take a closer look on the procedures outlined above.

7.3.3.1 Creating the Objects for the Remote Client

You need to define a proxy user and a network object for each remote client.

1. Start with creating the proxy user.

   Double-click on **Users** at the main panel of the firewall GUI. The User Administration panel comes up. Select **<NEW>** and click on the **Open...** button to access the Add User dialog (see Figure 80). Change the Non-Secure IP field under the Authentication section to **password**. Click on the **Password** tab and specify a password for this user, optionally set the password rules. Click on **OK** to finish the proxy user creation.



Identification	
Authority Level:	Proxy User
User Name:	tamas
User Full Name:	
Environment	
Secure Interface Shell:	/bin/restrict.sh
Non-Secure Interface Shell:	/bin/restrict.sh
Authentication	
Local Login:	deny
Secure FTP:	deny
Non-Secure FTP:	deny
Secure Telnet:	deny
Non-Secure Telnet:	deny
Secure IP:	deny
Non-Secure IP:	password
Secure Administration:	deny
Non-Secure Administration:	deny
Securenet Key:	
Session Control	
Warning Time:	20
Disconnect Time:	30

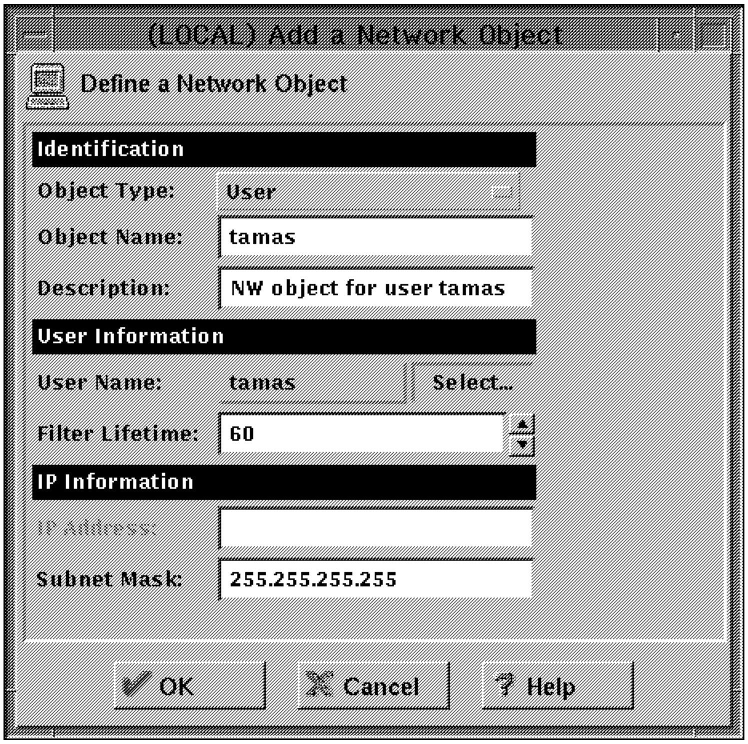
Figure 80. New Proxy User

Good to know

In Version 3.2.1 of the eNetwork Firewall for AIX, the Secure IP and Non-Secure IP fields shown in Figure 80 are replaced by the single Remote IP field. Also, the Password authentication method is called Firewall Password in V3.2.1.

2. Next, create a network object referring the proxy user just created. This is needed to be able to associate the dynamic tunnel with the user.

   Double-click on **Network Objects** at the main panel. Select **<NEW> Single** at the Network Objects Administration panel and click on **Open...** to access the panel shown in Figure 81. Fill in the fields as appropriate, then click **OK** to finish.



(LOCAL) Add a Network Object

Define a Network Object

Identification

Object Type: User

Object Name: tamas

Description: NW object for user tamas

User Information

User Name: tamas

Filter Lifetime: 60

IP Information

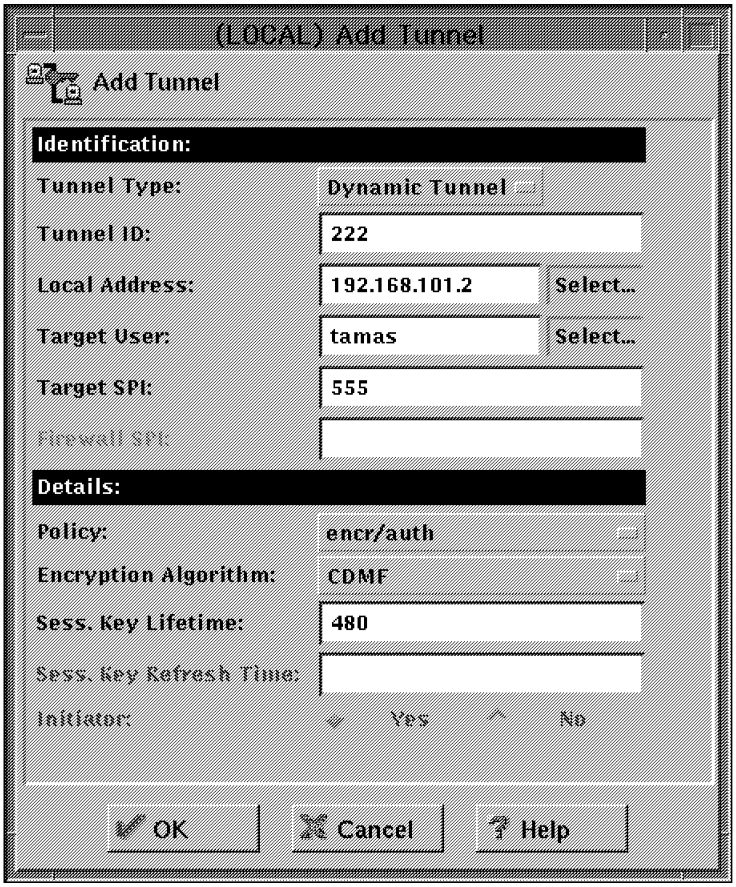
IP Address:

Subnet Mask: 255.255.255.255

Figure 81. Creating a Network Object for the Proxy User

7.3.3.2 Defining the Dynamic Tunnel

Now access the Virtual Private Network Administration panel to define the dynamic tunnel. Select **<NEW>** then click **Open...**. Select **Dynamic Tunnel** as the tunnel type and fill in the necessary fields. You are able to select the network object created for the proxy user by clicking on **Select...** near the Target User field. Do not type in the user name directly in this field. See Figure 82 on page 141 for an example.



Identification:

Tunnel Type: Dynamic Tunnel

Tunnel ID: 222

Local Address: 192.168.101.2

Target User: tamas

Target SPI: 555

Firewall SPI:

Details:

Policy: encr/auth

Encryption Algorithm: CDMF

Sess. Key Lifetime: 480

Sess. Key Refresh Time:

Initiator: ◆ Yes ◇ No

Figure 82. Defining a Dynamic Tunnel

Notes:

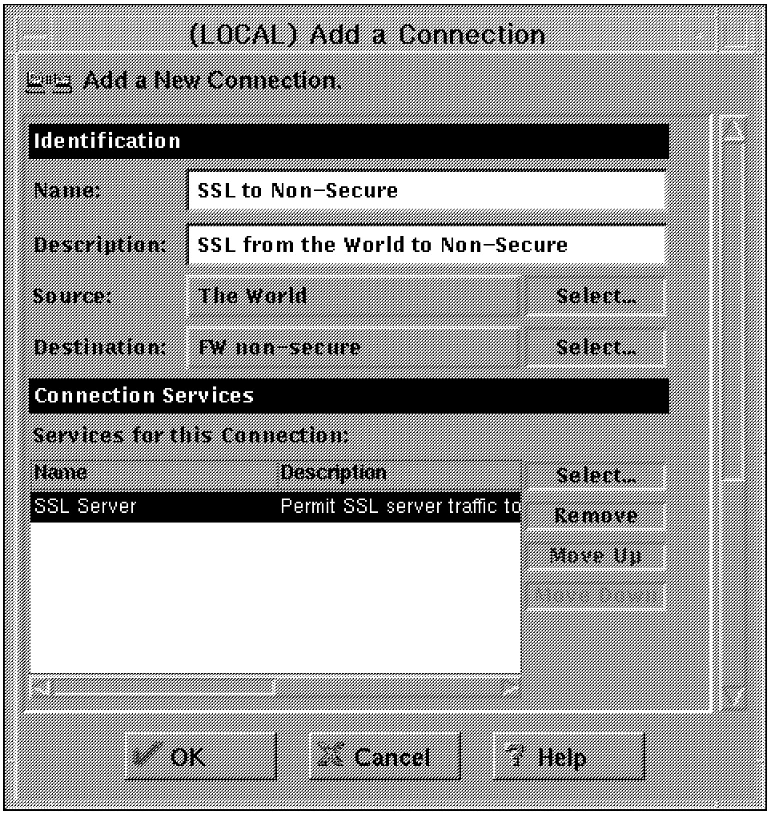
1. For a large number of tunnels it is useful to create a tunnel and SPI numbering standard to keep track of them more easily.

2. The Target SPI field means that the SPI specified here will be used by the target client when sending IPSec packets to the firewall. The Firewall SPI will be used by the firewall for outbound IPSec traffic to the client.

7.3.3.3 Creating the Connection for SSL

Now you have to allow SSL from all non-secure networks (network object The World) to the non-secure adapter. This is done by creating a new connection using the predefined service SSL Server.

Access the Connection Administration panel of the firewall GUI, select **New**, then click **Open...** and fill in the parameters needed for this connection as it is shown in Figure 83 on page 142.



(LOCAL) Add a Connection

Add a New Connection.

Identification

Name: SSL to Non-Secure

Description: SSL from the World to Non-Secure

Source: The World

Destination: FW non-secure

Connection Services

Services for this Connection:

Name	Description
SSL Server	Permit SSL server traffic to

Figure 83. Creating the SSL Connection

Finish this step by activating the connections so that the new filter rules will be put in operation.

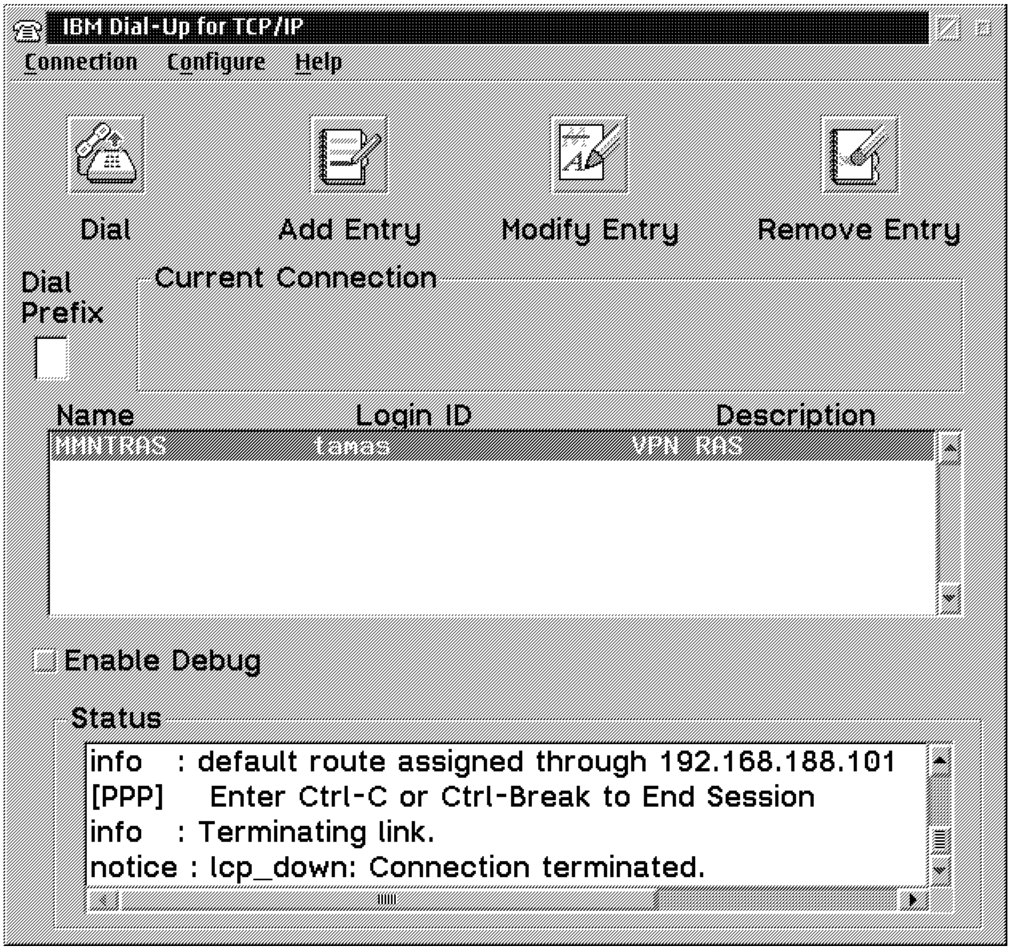
Note: The SSL connection creation step must be done only once; it serves all clients.

This concludes the firewall setup for dynamic tunnels. Now we turn to the remote client machine.

7.3.3.4 Configuring the Dial-In Connection to the ISP

If your ISP is the IBM Global Network, use the IBM Internet Dialer application. Otherwise, the Dial Other Internet Providers application should be used. In our test environment we used the latter. Your ISP should have provided the information necessary to configure the dial-in connection.

The dialer main window is shown in Figure 84 on page 143:



Name	Login ID	Description
MMNTRAS	tamas	VPN RAS

□ Enable Debug

```
info   : default route assigned through 192.168.188.101
[PPP]    Enter Ctrl-C or Ctrl-Break to End Session
info   : Terminating link.
notice : lcp_down: Connection terminated.
```

Figure 84. The OS/2 TCP/IP Dialer for Other Internet Providers

Click on **Add Entry** and fill in the fields in the Add Entries panel with the information from your ISP. Typically you will use a PPP connection. The Your IP Address, Destination IP Address and Network Mask fields do not have to be filled in, because the provider supplies these values upon PPP link establishment. On the last page of the panel you have to specify your modem parameters so that the dialer can access it correctly.

Good to know

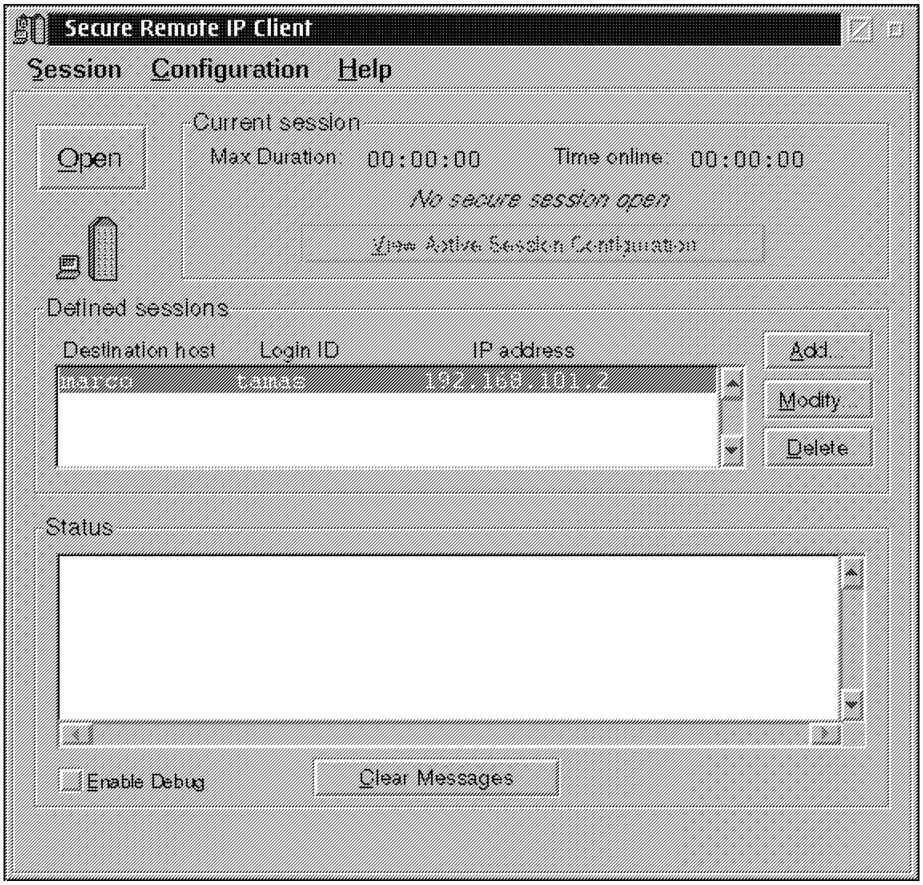
If the OS/2 system has simultaneous dial-up and LAN connections, the LAN default route will be replaced with the dial-up default route as long as the dial-up link is active. OS/2 TCP/IP by default creates a network route to the LAN IP network in order to maintain LAN connectivity in that case. If your LAN consists of multiple IP networks (not subnets), you need to add net route statements for each of them, using the TCP/IP Configuration Notebook, in order to access them while using a dial-up connection.

7.3.3.5 Defining the IPSec Connection to the Firewall

Once the dial-in connection to the ISP is operational, we can move on to define the IPSec connection to our firewall.

1. Open the **TCP/IP Configuration** folder and double-click the **Secure Remote IP Client Configuration**.

2. The Secure Remote IP Client window shown in Figure 85 comes up. Click **Add** to define a session (IPSec connection).



Secure Remote IP Client

Current session

Max Duration: 00:00:00 Time online: 00:00:00

No secure session open

Defined sessions

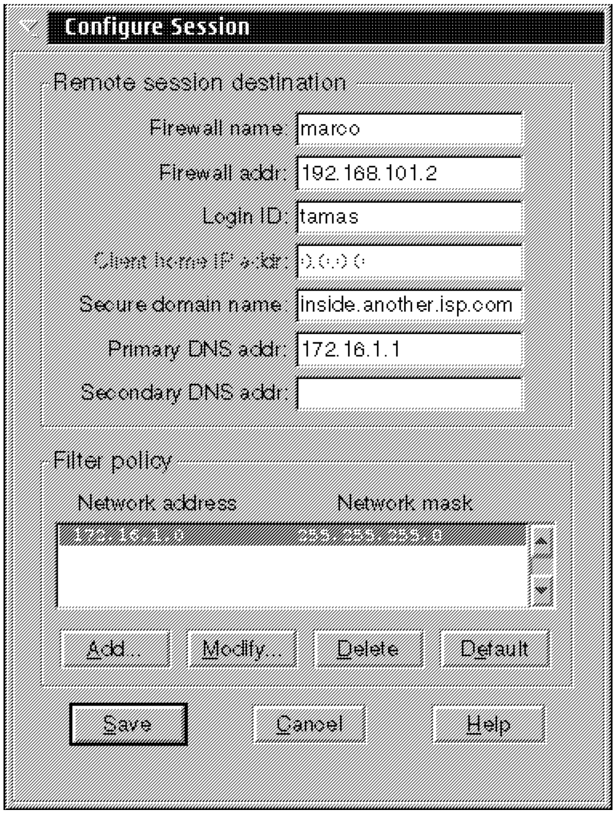
Destination host	Login ID	IP address
marco	tamas	192.168.101.2

Status

□ Enable Debug

Figure 85. OS/2 Secure Remote IP Client

3. At the Configure Session dialog you have to enter information about the remote firewall and intranet: firewall name and IP address, login ID, secure domain name and name server addresses. The firewall name does not need to be identical with the firewall′s hostname, it is used only locally as a session identifier. See Figure 86 on page 145 for an example.



Firewall name: marco

Firewall addr: 192.168.101.2

Login ID: tamas

Client home IP addr: 0.0.0.0

Secure domain name: inside.another.isp.com

Primary DNS addr: 172.16.1.1

Secondary DNS addr:

Network address	Network mask
172.16.1.0	255.255.255.0

Figure 86. Configure Session Dialog

Next, add one or more filter policy entries. These entries specify which traffic is to be sent through the tunnel. If you do not enter any, all traffic will be sent to the firewall through the tunnel and you probably will not be able to access Internet resources as long as the tunnel is active.

Click on **Add** and enter an IP network address and the associated network mask. Repeat the step for each subnet to be reached in the intranet.

You have now finished the session definition. Save it and you will be returned to the main window.

Good to know

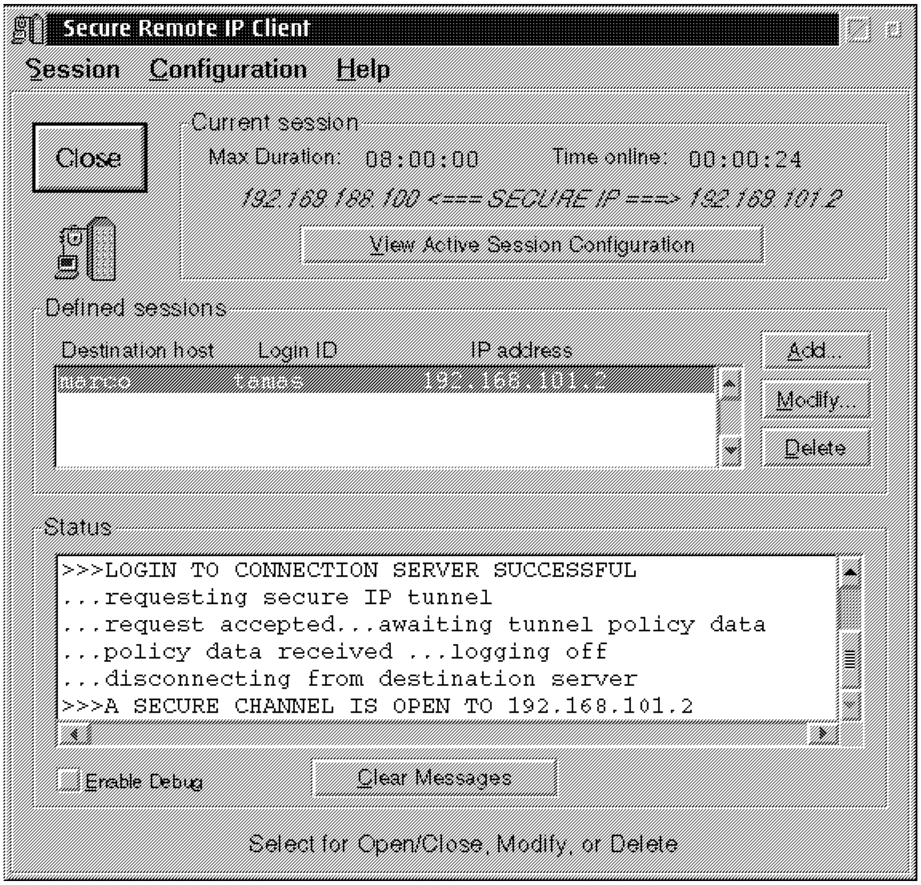
Because the OS/2 TCP/IP V4.1 IPSec Client, unlike the Windows 95 IPSec Client, is independent of the data link layer protocol, it can also be used in a LAN environment. For LAN usage the same setup procedure applies, only the dial-up connection setup is irrelevant.

7.4 Activating and Deactivating the Dynamic Tunnel

The dynamic tunnel activation is simple. At the firewall side there is nothing to do after the tunnel was defined and the SSL connection activated.

At the remote client establish the dial-in connection to your ISP, then open the Secure Remote IP Client Configuration window. The IPSec connection you defined earlier is listed. Select it and then click on **Open**. Specify your password. The client now starts the SSL control session to the firewall and retrieves the tunnel policy. You can see the tunnel activation progress in the lower textbox of the client.

After a while the IPSec connection is established and the client shows IPSec status information, like in the figure below:



Max Duration: 08:00:00 Time online: 00:00:24

192.169.188.100 <=== SECURE IP ===> 192.169.101.2

Destination host	Login ID	IP address
marco	tames	192.168.101.2

```
>>>LOGIN TO CONNECTION SERVER SUCCESSFUL
...requesting secure IP tunnel
...request accepted...awaiting tunnel policy data
...policy data received ...logging off
...disconnecting from destination server
>>>A SECURE CHANNEL IS OPEN TO 192.168.101.2
```

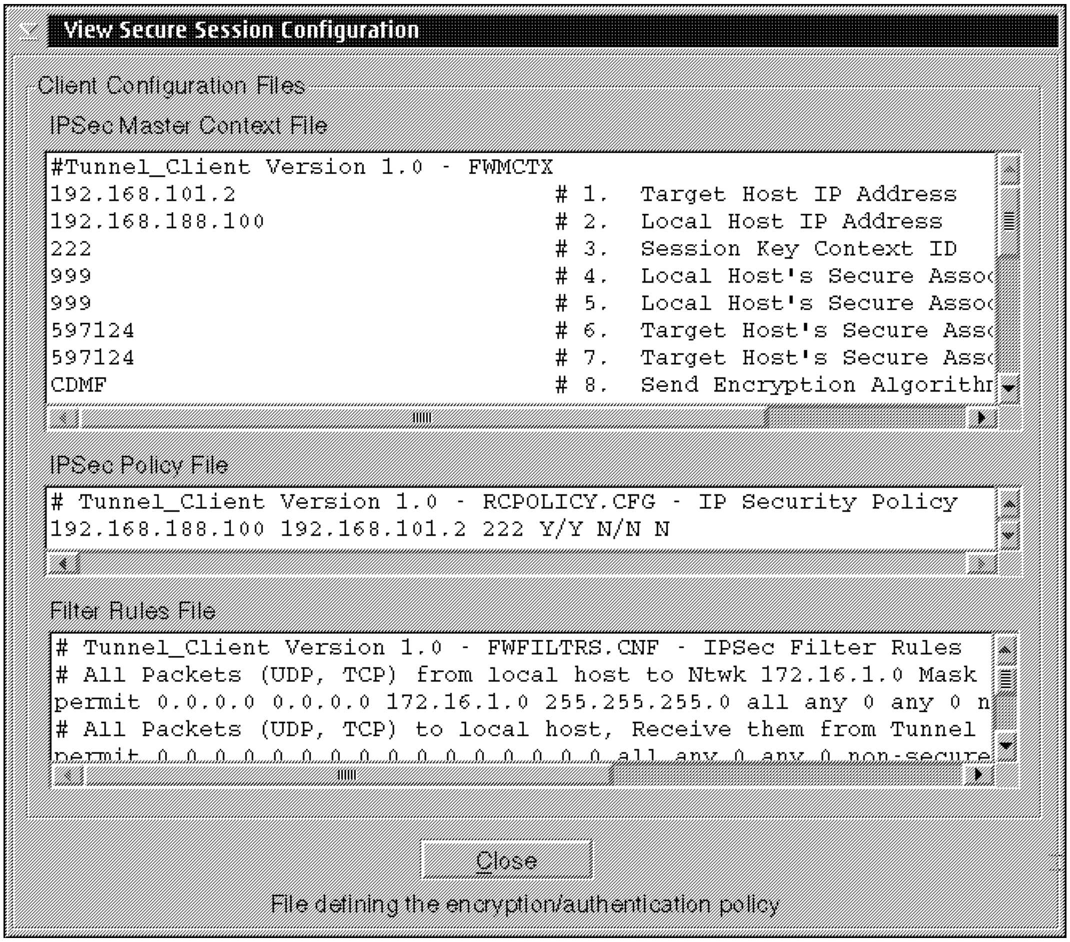
□ Enable Debug

Select for Open/Close, Modify, or Delete

Figure 87. An Active IPSec Connection

Note: While you can have multiple secure sessions defined, only one can be opened at any time.

By clicking on **View Active Session Configuration** you can take a look at the parameters of this session. See Figure 88 on page 147 for an illustration.



IPSec Master Context File

```
#Tunnel_Client Version 1.0 - FWMCTX
192.168.101.2                      # 1.  Target Host IP Address
192.168.188.100                    # 2.  Local Host IP Address
222                                # 3.  Session Key Context ID
999                                # 4.  Local Host's Secure Assoc
999                                # 5.  Local Host's Secure Assoc
597124                             # 6.  Target Host's Secure Ass
597124                             # 7.  Target Host's Secure Ass
CDMF                               # 8.  Send Encryption Algorith
```

IPSec Policy File

```
# Tunnel_Client Version 1.0 - RCPOLICY.CFG - IP Security Policy
192.168.188.100 192.168.101.2 222 Y/Y N/N N
```

Filter Rules File

```
# Tunnel_Client Version 1.0 - FWFILTRS.CNF - IPSec Filter Rules
# All Packets (UDP, TCP) from local host to Ntwk 172.16.1.0 Mask
permit 0.0.0.0 0.0.0.0 172.16.1.0 255.255.255.0 all any 0 any 0 n
# All Packets (UDP, TCP) to local host, Receive them from Tunnel
permit 0.0.0.0 0.0.0.0 0.0.0.0 0.0.0.0 all any 0 any 0 non-secure
```

File defining the encryption/authentication policy

Figure 88. IPSec Session Configuration

Examining the display one can see that the dynamic tunnel setup is similar to the IBM or manual tunnels. The IPSec Master Context File received from the firewall is analogous to the fwexpmctx file that is created when exporting an IBM tunnel. The firewall exports the dynamic tunnel definition upon the client′s request.

The tunnel is active for the time set by its Session Key Lifetime parameter at the firewall.

You can shut down the tunnel any time by selecting the active session and clicking on **Close**. This will remove the associated dynamic filter rules at the firewall. However, if the IP connectivity is lost without closing the tunnel (for example, due to inactivity timeout or power off) at the client, the firewall will not remove the dynamic filter rules until the session key lifetime expires. If you still want to close the tunnel, you first have to reestablish IP connectivity to the firewall. If you are using a dial-up connection, it does not matter if you get a different IP address from your ISP than the one you had when the tunnel was established.

At the firewall you can clear all dynamic filter rules thus shutting down all dynamic tunnels with the dfclr command. Selective dynamic tunnel shutdown is possible with the dfdel <username> <tunnelid> command. A listing of the dynamic filter rules can be obtained by issuing the dfdump command (see also 8.3.1, “Helpful Commands and Tools” on page 159).

Attention

dfclr, dfdel and dfdump are undocumented commands. They might be changed or missing in future IBM eNetwork Firewall for AIX releases without notice.

7.5 Testing the Dynamic Tunnel

If the firewall and the client are set up correctly and you are able to reach the intranet behind it with the IPSec connection active, then the dynamic tunnel should work.

You can test the dynamic tunnel at the firewall with the admin_test command. (see 8.2.2.1, “The admin_test Command” on page 153.)

Maybe the simplest way to see whether a dynamic tunnel is active or not is the dfdump command. It takes no parameters and lists all active dynamic filter rules, with user name information.

Since the filter rules associated with a dynamic tunnel are auto-generated and are always checked before the static rule base, the operation is very reliable. You cannot misconfigure it.

If you have problems getting the dynamic tunnel up and running, try 8.5, “OS/2 TCP/IP V4.1 IPSec Client” on page 164, where you can also find a description of the logging facility of the OS/2 TCP/IP V4.1 IPSec Client.

7.6 An Insight View to Dynamic Tunnels

In this section we describe in detail how a dynamic tunnel works by following the itinerary of an IP packet from the remote client to the destination host in the intranet.

When the remote client user opens a secure session, the GUI starts an SSL control session with the firewall, using TCP on port 4005. The SSL server application on the firewall authenticates the client based on user ID and password, sends the tunnel keys and policy and activates the dynamic tunnel and dynamic filter rules. After that, the SSL control session is terminated. This traffic is allowed by the SSL from the World connection that translates to the following static filter rules:

```
permit 0 0 0 0 tcp      any     0  eq 4005 both both both l=y
permit 0 0 0 0 tcp/ack  eq   4005 any    0 both both both l=y
```

Now the client can start the tunnel and activate its own filter rules that basically steer the corporate traffic into or out of the tunnel.

The dynamic filter rules activated by the SSL server application on the firewall are listed below. For easier reading we use the following notations:

FW: IP address of the firewall (non-secure interface)

CLI: IP address of the remote client

FF: 255.255.255.255

x: tunnel number

z: 4005

```
permit  FW FF CLI FF tcp eq  z any 0 non-secure local outbound l=y f=y t=0  (1)
permit CLI FF  FW FF esp any 0 any 0 non-secure local inbound  l=y f=y t=0  (2)
permit CLI FF  FW FF ah  any 0 any 0 non-secure local inbound  l=y f=y t=0  (3)
permit CLI FF   0  0 all any 0 any 0 non-secure both  inbound  l=y f=y t=x  (4)
permit   0  0 CLI FF all any 0 any 0 non-secure both  outbound l=y f=y t=x  (5)
permit CLI FF   0  0 all any 0 any 0 secure     route outbound l=y f=y t=0  (6)
permit   0  0 CLI FF all any 0 any 0 secure     route inbound  l=y f=y t=0  (7)
```

Notes:

1. Rule (1) permits the SSL server application to initiate a TCP connection to the specific remote client. Observe that the static filter rules do not allow this.
2. Rules (2) and (3) allow inbound IPSec traffic from the client. These are the ″green light″ for the encapsulated IP packets.
3. Rules (4) and (5) specify that client traffic should go through the tunnel x in both directions. Rule (4) is applied after the inbound IPSec packet was decapsulated. Rule (5) is applied before the outbound packet is encapsulated and actually instructs the firewall to do this encapsulation.
4. Rules (6) and (7) permit unrestricted bidirectional access to the secure network.
5. On outbound processing the filters are not checked after the IPSec encapsulation, that′s why there is no need for a permit rule that allows outbound IPSec packets. Rule (5) is sufficient.

Now let′s take a look on the process flow inside the firewall (see Figure 89).

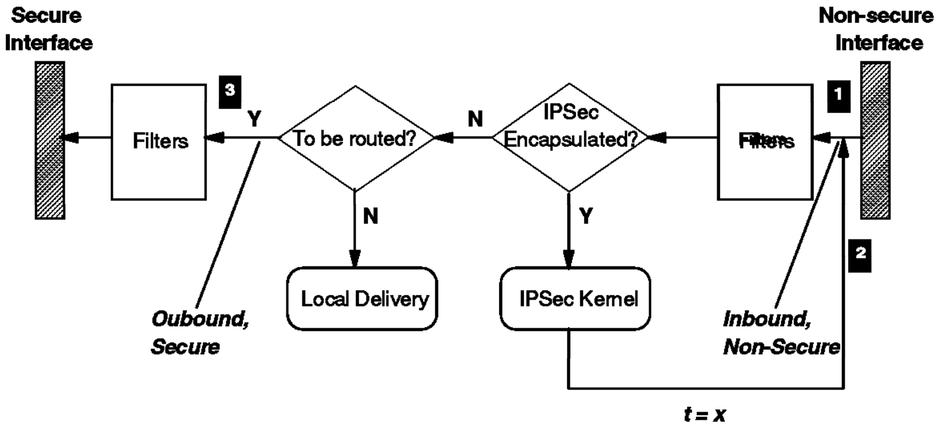


Figure 89. Inbound Packet Processing by the Firewall

Suppose that an IPSec packet arrives at the non-secure interface of the firewall, with the destination in the secure network. The tunnel policy is encr/auth, that is, the outer IPSec header is AH. Let′s denote dynamic filter rule r by DF(r).

1. The packet attributes are inbound, non-secure. As it is checked against the dynamic filter rules (remember, they are always checked first), a match is found with DF(3), so it passes to the IPSec decapsulation. Using the SPI, the

IPSec kernel retrieves the tunnel policy, strips off the IPSec headers, associates the tunnel ID x to the packet and sends it to the filters again.

2. Now DF(4) is matched and the packet proceeds to IP routing check.

3. The packet has to be routed, therefore the filters are checked again, but now the attributes of the packet are outbound, secure. There is a match with DF(6), so the packet is sent to the secure interface, which delivers it to the secure network.

The reverse flow is shown in the figure below.

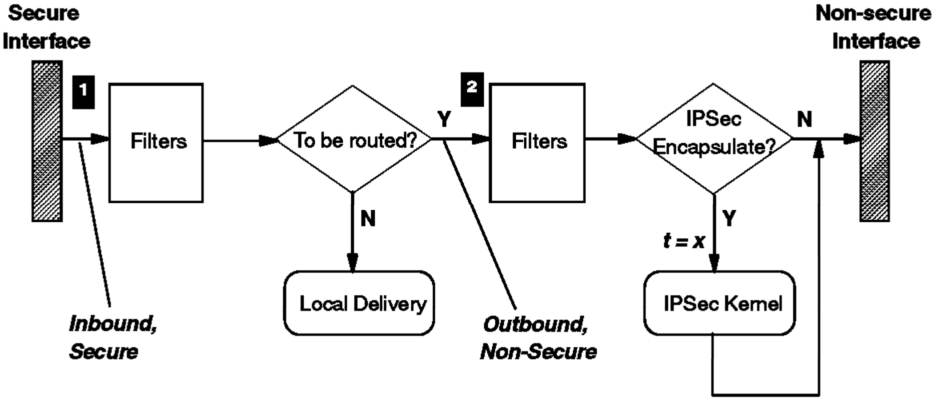


Figure 90. Outbound Packet Processing by the Firewall

A plain IP packet destined to the remote client arrives at the secure interface. Here is what happens:

1. The packet attributes are inbound, secure. At the first rules check DF(7) matches and the packet is accepted.

2. Since this packet has to be routed, the filters are checked again, but now the packet attributes are outbound, non-secure. Now DF(5) matches, and because it indicates a nonzero tunnel ID, the packet is sent to IPSec encapsulation. The IPSec kernel processes the packet based on the tunnel policy that it retrieved using the tunnel ID. The encapsulated packet is not sent to the filters again, it goes directly to the non-secure interface and travels to the client.

When the remote client closes the secure session, the GUI starts another SSL control session with the firewall. The same authentication takes place as in the case of session opening. Then the firewall deactivates the dynamic tunnel and removes the dynamic filter rules for the remote user. The client is notified and the SSL control session is terminated.

Chapter 8. Troubleshooting Your VPN

This chapter helps you in solving IPSec-related problems. We grouped this chapter in a general hints section that we think is valid for all products and information on the specific products. The more general hints of this chapter have been also partly included in Chapter 4, "IBM eNetwork VPN Solutions" on page 59 as long as we thought the information is important enough to justify the duplication.

8.1 General Hints

This section summarizes our general experiences and the lessons we have learned.

8.1.1 Double-Check the Field Entries

The most probable cause of a non-working tunnel are field entries that do not match the values entered on the partner system. Remember that your source addresses, SPIs and keys are reflected in the corresponding destination fields of your partners tunnel definition.

Some products, for instance, require a hexadecimal value for the SPI while others expect to receive a decimal number. In this case you have to manually convert the numbers before comparing it to your partners definition.

Of course both tunnel partners need to use the same protocols, header formats, transforms and policies.

8.1.2 Packet Filters

You have to take care that your VPN services go to the top of the filter file (for instance /etc/security/fwfilters.cfg on a eNetwork Firewall for AIX). Otherwise it might well be that other filter rules are evaluated before the VPN services permit or deny traffic that you intend to send through your tunnel.

You should understand the way of a packet described in 5.6, "How Does It Work?" on page 104; this will help you analyze problems.

8.1.3 Logging

The log files are among the most valuable tools in overcoming IPSec-related problems.

However, a logged packet containing a tunnel ID does not guarantee that the packet has been sent from the firewall to the other tunnel partner. It only tells you that the firewall code has sent the packet to the IPSec kernel. If the tunnel partners do not talk to each other, the IPSec kernel will not send the packet. You are able to see this if you trace the connection. You will see a logged packet in the log file but no corresponding packet in the IP trace.

If you see a log entry for the encrypted or authenticated packet leaving your system (non-secure outbound), then the tunnel packet has really been sent out. Unfortunately, some systems, such as the eNetwork Firewall for AIX, do not log this packet.

8.1.4 Nested Tunnels

If you work with nested tunnels, you should first try to get the tunnel between the firewalls to work. After that give the nested tunnel a try.

8.2 eNetwork Firewall for AIX

This section summarizes our experiences with this IBM eNetwork VPN solution.

8.2.1 Gotchas

The most common mistakes and misunderstandings that we came across are listed in this section.

8.2.1.1 AH Transform HMAC-MD5

From looking at some SMIT definition panels and the syntax of the fwtunnl command you could get the impression that HMAC MD5 is a supported authentication transform. Unfortunately this is currently not true. The only supported AH transform is keyed MD5. This will change later this year when the eNetwork Firewall for AIX gets the IPSec functionality currently only available in AIX V4.3.

8.2.1.2 Logging

The default logging action for the VPN related services has changed between V3.1.1.4 and V3.1.1.5. The value for logging used to be Yes and has changed to No. Therefore you manually need to override the default if you want to analyze tunnel traffic in the log file.

Remember from the general logging section above that that outbound AH and ESP packets are not logged. You cannot change this, so don′t get confused by this fact.

8.2.1.3 Tunnel Policy

With a policy of encr/auth data (as you would expect) is first encrypted and then authenticated. But take care. If you issue an `fwtunnl cmd=list` command to list your tunnel definition or if you look at the short policy name in the VPN section of the GUI the short name for encr/auth is ae (in contrary to the AIX V4.3 conventions - see 4.2.2, "AIX V4.3" on page 67). The developers decided that ae on the firewall stands for after encryption, meaning authentication after encryption which is the same as first encrypt and then authenticate.

8.2.1.4 SPI Naming

The naming conventions for SPIs on the panels are misleading. Trust the output of the admin_test and export files. They use the correct titles for the SPIs.

8.2.2 Helpful Commands and Tools

The following commands are helpful in diagnosing VPN problems on the firewall:

8.2.2.1 The admin_test Command

Admin_test will tell you the current status of a tunnel. It is undocumented and the syntax is as follows:

```
admin_test
```

After entering the command you have to type the source and destination IP addresses of the firewalls, then press Enter:

```
<sourceIP> <destIP>
```

Note: Always use the IP address of the local firewall first.

If key info is returned, a tunnel is up. In this case there is also additional information such as the start and end times for the current session key.

If a message `get_sess_key_ctx_from_cache() failed` is returned, the tunnel is not active.

One possible syntax variation is:

```
admin_test <filename>
```

where filename consists of one line containing the source and destination IP-address of the defined tunnel. An example of the output of the admin_test command is shown below:

```
A session key context :
his addr 9.24.105.171
my addr 9.24.105.170
context id 64 100
his esp said 0 0
his ah said 0 0
my esp said 1 1
my ah said 1 1
s enc alg 18
s_enc_ key len 8
s enc key :
        fc ac  e a1 b7 60 43 91

r enc alg 18
r enc  key len 8
r enc key :
        17 4c f6 67 1a 24 f2 43

s mac alg 5
s mac key len 16
s mac key :
        8a 9b a7 78 4a 62 ba 12
        1d 51 cc f7 51 24 f3 a3

r mac alg 5
r mac key len 16
r mac key :
        fa  a 8f 7f d0 d4 5b 6c
        23 35 9b 13 d3 7f 14 ba

start 895692757 Wed May 20 15:32:37 1998

end 895694557 Wed May 20 16:02:37 1998
```

The main value of this command is for IBM tunnels where it will really tell you whether the tunnel partners are able to communicate. If you use manual tunnels, the command will always give a status of a working tunnel, whether the partner is there or not. It is still quite useful even for manual tunnels, because all tunnel characteristics are shown.

There are three commands that will help you with the dynamic filter rules set by the firewall. Use them if you work with the OS/2 TCP/IP V4.1 IPSec Client and the Windows 95 IPSec Client.

8.2.2.2 The dfdump Command

The first is the dfdump command. It is undocumented but quite useful. On the firewall, typing dfdump in a window will show all dynamic filter rules in the kernel in stanza format. The firewall creates seven rules for each active connection. An example of the output of the dfdump command is shown below:

```
There are 14 dynamical filter rules:

The 1th rule is:
Action of the rule is: FLTR_PERMIT
The source addr.   is: 9.24.105.170
The source addr. mask: 255.255.255.255
The dest.  addr.   is: 192.168.100.125
The dest.  addr. mask: 255.255.255.255
Protocol of  rule  is: RC_FWPROTO_TCP
The source port    is: 4005
The source port op is: RC_FLTR_EQ
The dest.  port    is: 0
The dest.  port op is: FLTR_ANY
The adapter        is: FLTR_NONSECURE
The scope          is: FLTR_LOCAL
The direction      is: FLTR_OUTBOUND
The tunnel id      is: 0
The Enc            is: 0
The Mac            is: 0
The  flag          is: FRAGMENTS_YES
The inittime       is: 896046124
The Lifetime       is: 28800
The  UserId        is: root

The 2th rule is:
Action of the rule is: FLTR_PERMIT
The source addr.   is: 192.168.100.125
The source addr. mask: 255.255.255.255
The dest.  addr.   is: 9.24.105.170
The dest.  addr. mask: 255.255.255.255
Protocol of  rule  is: RC_FWPROTO_ESP
The source port    is: 0
The source port op is: RC_FLTR_ANY
The dest.  port    is: 0
The dest.  port op is: FLTR_ANY
The adapter        is: FLTR_NONSECURE
The scope          is: FLTR_LOCAL
The direction      is: FLTR_INBOUND
The tunnel id      is: 0
The Enc            is: 0
The Mac            is: 0
The  flag          is: FRAGMENTS_YES
The inittime       is: 896046124
The Lifetime       is: 28800
The  UserId        is: root

The 3th rule is:
Action of the rule is: FLTR_PERMIT
The source addr.   is: 192.168.100.125
The source addr. mask: 255.255.255.255
The dest.  addr.   is: 9.24.105.170
The dest.  addr. mask: 255.255.255.255
Protocol of  rule  is: RC_FWPROTO_AH
The source port    is: 0
The source port op is: RC_FLTR_ANY
The dest.  port    is: 0
The dest.  port op is: FLTR_ANY
The adapter        is: FLTR_NONSECURE
The scope          is: FLTR_LOCAL
The direction      is: FLTR_INBOUND
The tunnel id      is: 0
The Enc            is: 0
The Mac            is: 0
The  flag          is: FRAGMENTS_YES
The inittime       is: 896046124
The Lifetime       is: 28800
The  UserId        is: root

The 4th rule is:
Action of the rule is: FLTR_PERMIT
The source addr.   is: 192.168.100.125
The source addr. mask: 255.255.255.255
The dest.  addr.   is: 0.0.0.0
The dest.  addr. mask: 0.0.0.0
Protocol of  rule  is: RC_FWPROTO_ALL
The source port    is: 0
The source port op is: RC_FLTR_ANY
The dest.  port    is: 0
The dest.  port op is: FLTR_ANY
The adapter        is: FLTR_NONSECURE
The scope          is: FLTR_BOTH
The direction      is: FLTR_INBOUND
The tunnel id      is: 300
The Enc            is: 2
The Mac            is: 5
The  flag          is: FRAGMENTS_YES
The inittime       is: 896046124
The Lifetime       is: 28800
The  UserId        is: root

The 5th rule is:
Action of the rule is: FLTR_PERMIT
The source addr.   is: 0.0.0.0
The source addr. mask: 0.0.0.0
The dest.  addr.   is: 192.168.100.125
The dest.  addr. mask: 255.255.255.255
Protocol of  rule  is: RC_FWPROTO_ALL
The source port    is: 0
The source port op is: RC_FLTR_ANY
The dest.  port    is: 0
The dest.  port op is: FLTR_ANY
The adapter        is: FLTR_NONSECURE
The scope          is: FLTR_BOTH
The direction      is: FLTR_OUTBOUND
The tunnel id      is: 300
The Enc            is: 2
The Mac            is: 5
The  flag          is: FRAGMENTS_YES
The inittime       is: 896046124
The Lifetime       is: 28800
The  UserId        is: root

The 6th rule is:
Action of the rule is: FLTR_PERMIT
The source addr.   is: 192.168.100.125
The source addr. mask: 255.255.255.255
The dest.  addr.   is: 0.0.0.0
The dest.  addr. mask: 0.0.0.0
Protocol of  rule  is: RC_FWPROTO_ALL
The source port    is: 0
The source port op is: RC_FLTR_ANY
The dest.  port    is: 0
The dest.  port op is: FLTR_ANY
The adapter        is: FLTR_SECURE
The scope          is: FLTR_ROUTE
The direction      is: FLTR_OUTBOUND
The tunnel id      is: 0
The Enc            is: 0
The Mac            is: 0
The  flag          is: FRAGMENTS_YES
The inittime       is: 896046124
The Lifetime       is: 28800
The  UserId        is: root

The 7th rule is:
Action of the rule is: FLTR_PERMIT
The source addr.   is: 0.0.0.0
The source addr. mask: 0.0.0.0
The dest.  addr.   is: 192.168.100.125
The dest.  addr. mask: 255.255.255.255
Protocol of  rule  is: RC_FWPROTO_ALL
The source port    is: 0
The source port op is: RC_FLTR_ANY
The dest.  port    is: 0
The dest.  port op is: FLTR_ANY
The adapter        is: FLTR_SECURE
The scope          is: FLTR_ROUTE
The direction      is: FLTR_INBOUND
The tunnel id      is: 0
The Enc            is: 0
The Mac            is: 0
The  flag          is: FRAGMENTS_YES
The inittime       is: 896046124
The Lifetime       is: 28800
The  UserId        is: root

The 8th rule is:
Action of the rule is: FLTR_PERMIT
The source addr.   is: 9.24.105.170
The source addr. mask: 255.255.255.255
The dest.  addr.   is: 192.168.100.127
The dest.  addr. mask: 255.255.255.255
Protocol of  rule  is: RC_FWPROTO_TCP
The source port    is: 4005
The source port op is: RC_FLTR_EQ
The dest.  port    is: 0
The dest.  port op is: FLTR_ANY
The adapter        is: FLTR_NONSECURE
The scope          is: FLTR_LOCAL
The direction      is: FLTR_OUTBOUND
The tunnel id      is: 0
The Enc            is: 0
The Mac            is: 0
The  flag          is: FRAGMENTS_YES
The inittime       is: 896051790
The Lifetime       is: 28800
The  UserId        is: root

The 9th rule is:
Action of the rule is: FLTR_PERMIT
The source addr.   is: 192.168.100.127
The source addr. mask: 255.255.255.255
The dest.  addr.   is: 9.24.105.170
The dest.  addr. mask: 255.255.255.255
Protocol of  rule  is: RC_FWPROTO_ESP
The source port    is: 0
The source port op is: RC_FLTR_ANY
The dest.  port    is: 0
The dest.  port op is: FLTR_ANY
The adapter        is: FLTR_NONSECURE
The scope          is: FLTR_LOCAL
The direction      is: FLTR_INBOUND
The tunnel id      is: 0
The Enc            is: 0
The Mac            is: 0
The  flag          is: FRAGMENTS_YES
The inittime       is: 896051790
The Lifetime       is: 28800
The  UserId        is: root

The 10th rule is:
Action of the rule is: FLTR_PERMIT
The source addr.   is: 192.168.100.127
The source addr. mask: 255.255.255.255
The dest.  addr.   is: 9.24.105.170
The dest.  addr. mask: 255.255.255.255
Protocol of  rule  is: RC_FWPROTO_AH
The source port    is: 0
The source port op is: RC_FLTR_ANY
The dest.  port    is: 0
The dest.  port op is: FLTR_ANY
The adapter        is: FLTR_NONSECURE
The scope          is: FLTR_LOCAL
The direction      is: FLTR_INBOUND
The tunnel id      is: 0
The Enc            is: 0
The Mac            is: 0
The  flag          is: FRAGMENTS_YES
The inittime       is: 896051790
The Lifetime       is: 28800
The  UserId        is: root

The 11th rule is:
Action of the rule is: FLTR_PERMIT
The source addr.   is: 192.168.100.127
The source addr. mask: 255.255.255.255
The dest.  addr.   is: 0.0.0.0
The dest.  addr. mask: 0.0.0.0
Protocol of  rule  is: RC_FWPROTO_ALL
The source port    is: 0
The source port op is: RC_FLTR_ANY
The dest.  port    is: 0
The dest.  port op is: FLTR_ANY
The adapter        is: FLTR_NONSECURE
The scope          is: FLTR_BOTH
The direction      is: FLTR_INBOUND
The tunnel id      is: 300
The Enc            is: 2
The Mac            is: 5
The  flag          is: FRAGMENTS_YES
The inittime       is: 896051790
The Lifetime       is: 28800
The  UserId        is: root

The 12th rule is:
Action of the rule is: FLTR_PERMIT
The source addr.   is: 0.0.0.0
The source addr. mask: 0.0.0.0
The dest.  addr.   is: 192.168.100.127
The dest.  addr. mask: 255.255.255.255
Protocol of  rule  is: RC_FWPROTO_ALL
The source port    is: 0
The source port op is: RC_FLTR_ANY
The dest.  port    is: 0
The dest.  port op is: FLTR_ANY
The adapter        is: FLTR_NONSECURE
The scope          is: FLTR_BOTH
The direction      is: FLTR_OUTBOUND
The tunnel id      is: 300
The Enc            is: 2
The Mac            is: 5
The  flag          is: FRAGMENTS_YES
The inittime       is: 896051790
The Lifetime       is: 28800
The  UserId        is: root

The 13th rule is:
Action of the rule is: FLTR_PERMIT
The source addr.   is: 192.168.100.127
The source addr. mask: 255.255.255.255
The dest.  addr.   is: 0.0.0.0
The dest.  addr. mask: 0.0.0.0
Protocol of  rule  is: RC_FWPROTO_ALL
The source port    is: 0
The source port op is: RC_FLTR_ANY
The dest.  port    is: 0
The dest.  port op is: FLTR_ANY
The adapter        is: FLTR_SECURE
The scope          is: FLTR_ROUTE
The direction      is: FLTR_OUTBOUND
The tunnel id      is: 0
The Enc            is: 0
The Mac            is: 0
The  flag          is: FRAGMENTS_YES
The inittime       is: 896051790
The Lifetime       is: 28800
The  UserId        is: root

The 14th rule is:
Action of the rule is: FLTR_PERMIT
The source addr.   is: 0.0.0.0
The source addr. mask: 0.0.0.0
The dest.  addr.   is: 192.168.100.127
The dest.  addr. mask: 255.255.255.255
Protocol of  rule  is: RC_FWPROTO_ALL
The source port    is: 0
The source port op is: RC_FLTR_ANY
The dest.  port    is: 0
The dest.  port op is: FLTR_ANY
The adapter        is: FLTR_SECURE
The scope          is: FLTR_ROUTE
The direction      is: FLTR_INBOUND
The tunnel id      is: 0
The Enc            is: 0
The Mac            is: 0
The  flag          is: FRAGMENTS_YES
The inittime       is: 896051790
The Lifetime       is: 28800
The  UserId        is: root
```

8.2.2.3 The dfdel Command

Selective deactivation of dynamic tunnels can be done by using the dfdel command along with the username and tunnel ID of the specific tunnel that you want to shut down.

8.2.2.4 The dfclr Command

The dfclr command will remove all dynamic filter rules from the kernel.

8.3 AIX V4.3

This section summarizes our experiences with this IBM eNetwork VPN solution.

8.3.1 Helpful Commands and Tools

The following commands are very helpful when working with virtual private networks on AIX V4.3. Equivalent functionality (with the exception of the ipsecstat command) is also available via SMIT.

- lstun: Lists your current tunnel definitions and status
- lsfilt: Lists filter definitions
- ipsecstat: Useful overall status and statistic figures
- imptun: Import of tunnels defined on other systems
- exptun: Allows the export of your tunnel definitions

All VPN commands require root access to run. For the creation and handling of the tunnel and filter definitions we recommend to use SMIT. In addition, SMIT also offers an IPSec trace facility, available within the advanced IP security configuration panels. Within standard AIX logging the facility local4 handles all IPSec-related events.

8.3.2 Problem Determination

This section includes some hints and tips that may assist you when you encounter a problem. We recommend that you set up logging from the start; log files are very useful in determining what is going on with the filters and tunnels.

8.3.2.1 Case 1

Error:

Issuing a mktun command results in the following error:

```
insert_tun_man4(): write failed : The requested resource is busy.
```

Problem:

The tunnel you requested to activate is already active or you have colliding SPI values.

How to fix:

Issue the rmtun command to deactivate, then issue the mktun command to activate. Check to see if the SPI values for the failing tunnel match any other active tunnel. Each tunnel should have its own unique SPI values.

8.3.2.2 Case 2

Error:

Issuing a mktun command results in the following error:

```
Device ipsec_v4 is in Defined status.
Tunnel activation for IPv4 not performed.
```

Problem:

You have not made the ipsec device available.

How to fix:

Issue the following command: mkdev -l ipsec -t 4. You may have to change the -t option to 6 if you are getting the same error for v6 tunnel activation. The devices must be in available state. To check the ipsec device state, issue the following command: lsdev -Cc ipsec.

8.3.2.3 Case 3

Error:

Issuing a chfilt command results in the following error:

```
Cannot modify the first rule.
```

or

```
Cannot modify a pre_defined filter rule.
```

Problem:

You are not allowed to modify these filter rules. However, you may change whether they log or not.

How to fix:

If you want these rules to log, just issue the command: chfilt -v (4 or 6) -n (filter number) -l y. If you want to set up the default rules to pass AH or ESP packets to specific hosts only, then you may prevent the auto-generation of rules by using the -g parameter with the gentun command. Then you may add in the same rules for the AH and ESP packets with the specific host′s IP address for source and the partner host′s IP address for destination. Make sure these rules are placed before the actual tunnel traffic rules.

8.3.2.4 Case 4

Error:

Issuing a gentun command results in the following error:

```
Invalid Source IP address
```

Problem:

You have not entered a valid IP address for the source address.

How to fix:

For IPv4 tunnels, please check to see that you have entered an available IPv4 address for the local machine. You cannot use host names for the source when generating tunnels; you may only use host names for the destination. For IPv6 tunnels, please check to see that you entered an available IPv6 address. If you type netstat -in and no IPv6 addresses exist, run /usr/sbin/autoconf6 (interface) for a link local auto-generated address (using mac address) or use ifconfig to manually assign an address.

8.3.2.5 Case 5

Error:

Issuing a mktun command results in the following error:

```
insert_tun_man4(): write failed :
A system call received a parameter that is not valid.
```

Problem:

Tunnel generation occurred with invalid ESP and AH combination or without the use of the new header format when necessary.

How to fix:

Check to see what authentication algorithms are in use by the particular tunnel in question. Remember that the HMAC MD5 and HMAC SHA algorithms require the new header format. The new header format can be changed using SMIT or the -z parameter with the chtun command. Also remember that DES CBC 4 cannot be used with the new header format.

8.3.3 Logging

Below is a sample log file containing traffic entries and other IPSec log entries:

Note: Some lines had to be wrapped in order to fit on the page.

```
1.  Apr 27 08:08:40 host1 : Filter logging daemon ipsec_logd (level 2.20)
    initialized at 08:08:40 on 04/27/98
2.  Apr 27 08:08:46 host1 : mkfilt: Status of packet logging set to Start at
    08:08:46 on 04/27/98
3.  Apr 27 08:08:47 host1 : mktun: IBM tunnel 1, 9.3.97.244, 9.3.97.130
    activated.
4.  Apr 27 08:08:47 host1 skeyd: Inserted new context for tunnel ID 1 local SPI:
    1336 remote SPI: 1336 .
5.  Apr 27 08:08:47 host1 : mkfilt: #:1 permit 0.0.0.0 0.0.0.0 0.0.0.0 0.0.0.0
    udp eq 4001 eq 4001 both both l=n f=y t=0 e= a=
6.  Apr 27 08:08:47 host1 : mkfilt: #:2 permit 0.0.0.0 0.0.0.0 0.0.0.0 0.0.0.0
    ah any 0 any 0 both both l=n f=y t=0 e= a=
7.  Apr 27 08:08:47 host1 : mkfilt: #:3 permit 0.0.0.0 0.0.0.0 0.0.0.0 0.0.0.0
```

```
    esp any 0 any 0 both both l=n f=y t=0 e= a=
8.  Apr 27 08:08:47 host1 : mkfilt: #:4 permit 10.0.0.1 255.255.255.255 10.0.0.2
    255.255.255.255 icmp any 0 any 0 local outbound l=y f=y t=1 e= a=
9.  Apr 27 08:08:47 host1 : mkfilt: #:4 permit 10.0.0.2 255.255.255.255 10.0.0.1
    255.255.255.255 icmp any 0 any 0 local inbound l=y f=y t=1 e= a=
10. Apr 27 08:08:47 host1 : mkfilt: #:6 permit 0.0.0.0 0.0.0.0 0.0.0.0 0.0.0.0
    all any 0 any 0 both both l=y f=y t=0 e= a=
11. Apr 27 08:08:47 host1 : mkfilt: Filter support (level 1.00) initialized at
    08:08:47 on 04/27/98
12. Apr 27 08:08:48 host1 : #:6 R

 o:10.0.0.1 s:10.0.0.1 d:10.0.0.20
    p:udp sp:3327 dp:53 r:l a:n f:n T:0 e:n l:67
13. Apr 27 08:08:48 host1 : #:6 R:p i:10.0.0.1
    s:10.0.0.20 d:10.0.0.1 p:udp sp:53 dp:3327 r:l a:n f:n T:0 e:n l:133
14. Apr 27 08:08:48 host1 : #:6 R:p i:10.0.0.1
    s:10.0.0.15 d:10.0.0.1 p:tcp sp:4649 dp:23 r:l a:n f:n T:0 e:n l:43
15. Apr 27 08:08:48 host1 : #:6 R:p o:10.0.0.1
    s:10.0.0.1 d:10.0.0.15 p:tcp sp:23 dp:4649 r:l a:n f:n T:0 e:n l:41
16. Apr 27 08:08:48 host1 : #:6 R:p i:10.0.0.1
    s:10.0.0.15 d:10.0.0.1 p:tcp sp:4649 dp:23 r:l a:n f:n T:0 e:n l:40
17. Apr 27 08:08:51 host1 : #:4 R:p o:10.0.0.1
    s:10.0.0.1 d:10.0.0.2 p:icmp t:8 c:0 r:l a:n f:n T:1 e:n l:84
18. Apr 27 08:08:51 host1 : #:5 R:p i:10.0.0.1
    s:10.0.0.2 d:10.0.0.1 p:icmp t:0 c:0 r:l a:n f:n T:1 e:n l:84
19. Apr 27 08:08:52 host1 : #:4 R:p o:10.0.0.1
    s:10.0.0.1 d:10.0.0.2 p:icmp t:8 c:0 r:l a:n f:n T:1 e:n l:84
20. Apr 27 08:08:52 host1 : #:5 R:p i:10.0.0.1
    s:10.0.0.2 d:10.0.0.1 p:icmp t:0 c:0 r:l a:n f:n T:1 e:n l:84
21. Apr 27 08:32:27 host1 : Filter logging daemon terminating at 08:32:27 on
    04/27/98
```

Following are explanations to the points illustrated above:

1 Filter logging daemon activated.

2 Filter packet logging set to on with mkfilt -g start.

3 IBM tunnel activation, showing tunnel ID, source address, destination address, and time stamp.

4 The skeyd daemon inserted the tunnel context, meaning that the IBM tunnel is ready for traffic.

5-10 Filters have been activated; logging shows all loaded filter rules.

11 Message showing activation of filters.

12-13 These entries show a DNS lookup for a host.

14-16 These entries show a partial telnet connection.

17-20 These entries show two pings.

21 Filter logging daemon shutting down.

The fields in the log entries are abbreviated to reduce DASD space requirements. They have the following meanings:

The rule number that caused this packet to be logged.

R Rule type. p indicates permit; d indicates deny.

i/o Direction the packet was travelling when it was intercepted by the filter support code. Identifies IP address of the adapter associated with the packet. For inbound (i) packets, this is the adapter that the packet arrived on; for outbound (o) packets this is the adapter that the IP layer has determined should handle the transmission of the packet.

s Specifies the IP address of the sender of the packet (extracted from the IP header).

d Specifies the IP address of the intended recipient of the packet (extracted from the IP header).

p Specifies the high-level protocol that was used to create the message in the data portion of the packet. May be a number or name, for example: udp, icmp, tcp, tcp/ack, ospf, pip, esp, ah and all.

sp/t Specifies the protocol port number associated with the sender of the packet (extracted from the TCP/UDP header). When the protocol is ICMP or OSPF, this field is replaced with t which specifies the IP type. For more information, refer to your TCP/IP documentation.

dp/c Specifies the protocol port number associated with the intended recipient of the packet (extracted from the TCP/UDP header). When the protocol is ICMP, this field is replaced with c which specifies the IP code. For more information, refer to your TCP/IP documentation.

- Specifies that no information is available.

r Indicates whether or not the packet had any local affiliation. Can be f for forwarded packets, l for local packets, o for outgoing, b for both.

l Specifies the length of a particular packet in bytes.

f Identifies if the packet is a fragment.

T Indicates the tunnel ID.

i Specifies what interface the packet came in on.

8.4 Windows 95 IPSec Client

This section describes how we resolved the few problems we had installing Windows 95 IPSec Client.

8.4.1 Cannot Create Dial-Up Networking Entry

Installing Windows 95 IPSec Client after a failed attempt can sometimes result in an IBM IbmIsdn Software Adapter that is named slightly differently from what pppsec.exe expects. A successful install of pppsec.exe will create a dial-up networking connection for you. As it doesn′t recognize the slight name difference it will not work and give you an error.

To work around this you can manually create your own IBM IPSec dial-up connection using Windows 95 Dial-Up Networking Accessory. When selecting a device you should see:

```
IbmIsdn-Line01 Software Adapter
```

The number after Line might be slightly different.

Note: If the IbmIsdn entry cannot be seen when you select a device while attempting to make a new connection, then you may have a driver problem.

To correct this we completed the following steps:

1. Removed the TCPIP protocol from the network configuration in Windows 95.
2. Removed the IbmIsdn Software Network Adapter.
3. Reinstalled Microsoft ISDN Accelerator Pack 1.1.
4. Reinstalled the IbmIsdn Software Network Adapter.

When prompted by Windows 95 if we wanted to install drivers, we always kept the latest drivers. The most significant driver seemed to be ndis.vxd 4.1.1010. We accidentally replaced this with an older version and this caused the IbmIsdn Software Network Adapter to vanish.

8.4.2 Microsoft Dial-Up Networking 1.2

We tried to install Microsoft Dial-Up Networking 1.2 but it failed. Despite numerous attempts we could not get the IBM IbmIsdn Software Adapter device to become visible to the Create Dial-Up Adapter screen. The driver was visible to the System Devices view but that was it. We removed Microsoft Dial-Up Networking 1.2 and replaced the ndis.vxd and other drivers with older versions during the install of Microsoft ISDN Accelerator Pack 1.1.

8.4.3 SSL Error: Cannot Initiate Control Session to Firewall

Every attempt to start an IPSec Tunnel resulted in the following error:

```
SSL error
Cannot initiate control session to firewall.
Contact your firewall administrator.
SSL Init error code: 1
```

This was because the firewall was not running the process /usr/sbin/sslrctd. We had forgotten to modify /etc/security/rcsfile.cfg to point to our keyfile.kyr. This caused the sslrctd process to fail after a few minutes. We corrected the sslfile entry in rcsfile.cfg and ran sslrctd manually.

8.5 OS/2 TCP/IP V4.1 IPSec Client

This section describes some the troubleshooting features of the OS/2 TCP/IP V4.1 IPSec Client.

8.5.1 Problem Determination

Here are a few things that should be checked for the purpose of basic problem determination:

1. Take a look at the CONFIG.SYS file and make sure that the required device drivers IPSEC.SYS, CDMF.SYS and MD5.SYS are loaded.

2. Determine the level of eNetwork Firewall for AIX that you′re trying to connect to. You need a fix if it is V3.2 or V3.2.1.

3. Make sure that all definitions at the firewall are correct: proxy user, network object for proxy user, SSL connection, dynamic tunnel definition. (Remember that only CDMF is supported as encryption transform.)

4. Make sure that the SSL server (sslrctd) is active at the firewall.

5. Check the **Enable Debug** box in the Secure Remote IP Client window and try to open the tunnel again. Additional messages will be displayed in the Status window and will also be logged to the MPTNETCDEBUG.VPN file.

8.5.2 Logging

If the IPSec and packet filter syslog deamon (FSSD) is running, logs will be built in the MPTNETC directory. The FWLSLOG command is used to read the log and format/filter messages. The location of these files can be changed by setting the environment variable FWLOGS. An FWERROR.ERR file is generated in the MPTNETC directory to contain error messages that occurred while logging or during set up. The location of this file can be changed by setting the environment variable FWERROR.

Logging can be configured to limit the number and severity of messages logged. To set up logging, follow these steps:

1. Create a file FWLOG.CNF in the MPTNETC directory. This file indicates the level of logging to be done. The messages logged will be those equal to or greater than the level indicated in FWLOG.CNF. Level can be one of the following:

   - 10: Debug
   - 20: Informational
   - 30: Warning
   - 40: Errors
   - 50: Alert

   As an example, the contents of FWLOG.CNF:

   ```
   level=30
   ```

   would log all warning, error and alert messages.

2. Start the FSSD daemon by entering `fssd` on an OS/2 command line. This will create a log file in the MPTNETC directory and name it FW<DATE>, where <DATE> is the current date, for instance FW0529.

3. Start an IPSec tunnel to obtain the packet filter file from the firewall.

4. Edit the packet filter file, ETCSECURITYFWFILTRS.CNF, and append `l=y` to each filter rule for which you want packets to be logged.

5. Update the active filter rules by entering `cfgfilt -u-d`. You can ignore the message about a missing file.

6. Use the FWLSLOG command to access the current log. You cannot access it otherwise because it is locked by FSSD while it is running.

The following examples show two messages from the IPSec device driver that are logged in any case:

1.

   ```
   FW_LOG_02090 ICA1040w ICA1041i:
   Context specification deleted for tunnel: <tunnel ID>
   ```

   The tunnel context for the listed tunnel ID is no longer operational. It has either timed out or was deleted. Remember that a tunnel context is deleted whenever you close an IPSec connection.

2.

   ```
   FW_LOG_02200 : Invalid ipsec package:
   s:<source IP addr> d:<destination IP addr> protocol:<protocol> spi:<SPI>
   ```

   The device driver received an IPSec packet for which there was no context specification, or there was an authentication or decryption error.

8.6 Tracing an IPSec Connection

It is often useful to actually see the packets that make up an IPSec connection. You do not have to have specialized network analyzer hardware to do that. Most TCP/IP implementations have a tracing capability, either in the form of commands, for instance iptrace and ipreport on AIX, or iptrace and ipformat on OS/2, or GUI applications, such as IPTrace from the eNetwork Communications Suite and Windows NT Network Monitor.

Apart from IPTrace of the eNetwork Communications Suite and iptrace on OS/2 TCP/IP V4.1, common IP trace tools do not yet recognize IPSec protocols. However, you can look for the protocol numbers 50 (0x32) for AH and 51 (0x33) for ESP in the IP headers. In the case of nested protocols, you will not be able to easily analyze the encapsulated packet. You can recognize for example ESP encapsulated in AH by identifying protocol number 51 in the outer header and by looking for garbled payload data. The tools mentioned do not give a field-by-field breakdown of the inner headers, but if they are not encrypted, you could do it manually.

A more versatile tool is the Windows NT Network Monitor. It has quite sophisticated analyzer features, not limited to IP. With the version of Network Monitor that comes with Microsoft SMS, you can even capture packets of arbitrary destination and source addresses. However, the current version of Network Monitor does not recognize IPSec protocols either.

Notes:

1. Network Monitor will not work in capture mode unless your network adapter and driver support *promiscuous mode*; that is, it can receive LAN frames for any destination, not just those addressed to itself. For example, the IBM LANStreamer family of adapters can be put in promiscuous mode.
2. Network Monitor actually captures LAN data link frames, not just upper layer packets. The data link headers inserted before the IP packets are also shown.

We have used Network Monitor throughout the following sections for convenience as well as for illustration purposes, even though the NT system itself did not have any IPSec capabilities.

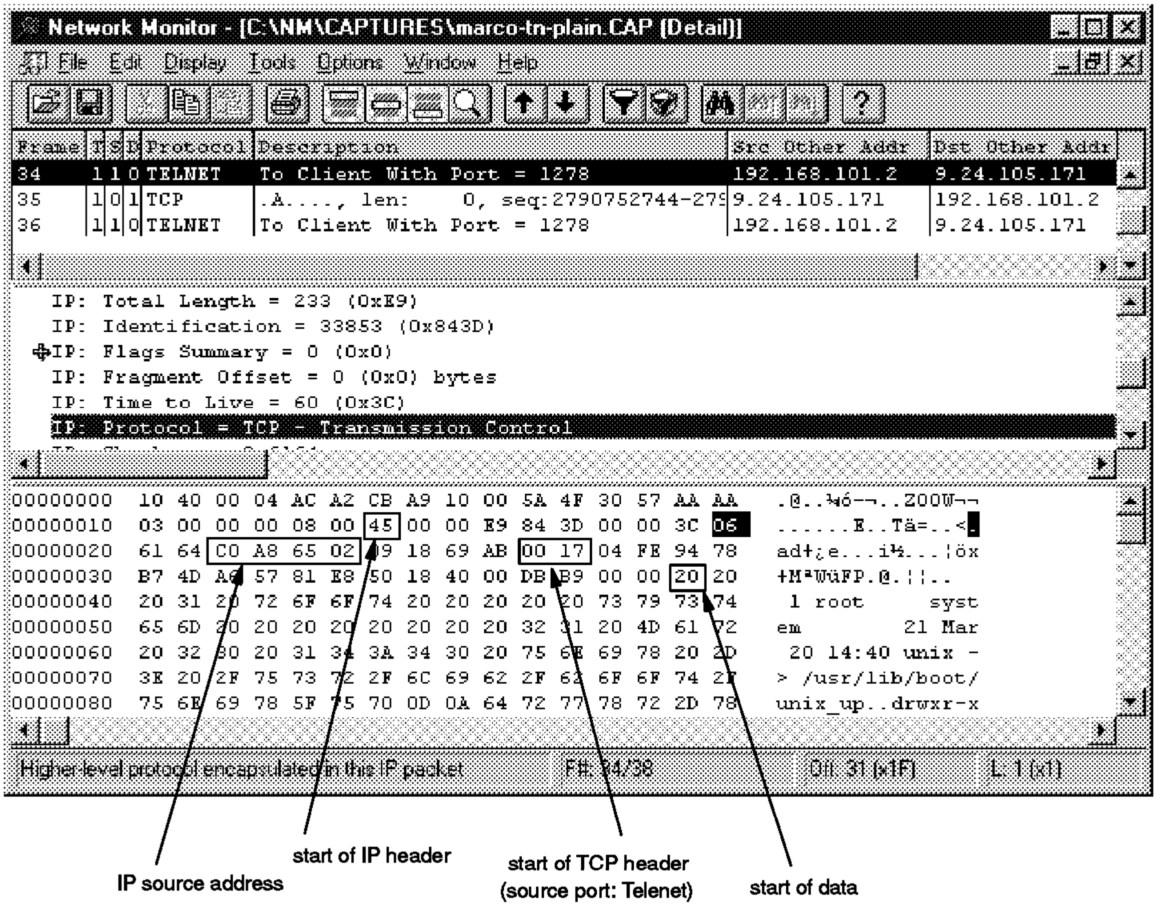
8.6.1 Trace Examples

Below we give one example of plain TCP/IP traffic and three examples of captured IPSec traffic. We used Network Monitor to verify the operation of a firewall-firewall IPSec tunnel in the scenario described in Chapter 5, “Branch Office Connection Scenario” on page 81. Follow the steps below to trace an IPSec connection:

1. Plug in the Network Monitor machine in a subnet that is crossed by the IPSec tunnel to be traced.
2. Define a filter that captures only IP traffic between the tunnel endpoints. See the Network Monitor documentation and help on how to do this.
3. Start the capture.
4. Generate some known traffic that goes through the tunnel; for example, open a telnet session, issue commands that list something, etc.
5. Stop the capture and save it. Then open the saved file for analysis.

8.6.1.1 Example 1: Plain TCP/IP Connection

Figure 91 shows the trace of a plain TCP/IP connection between the firewalls marco (192.168.101.2) and rs600024 (9.24.105.171). Note that the traffic is in plaintext and Network Monitor was able to determine that this is a telnet session and gave details on the TCP header. The telnet password would clearly be seen in such a trace.



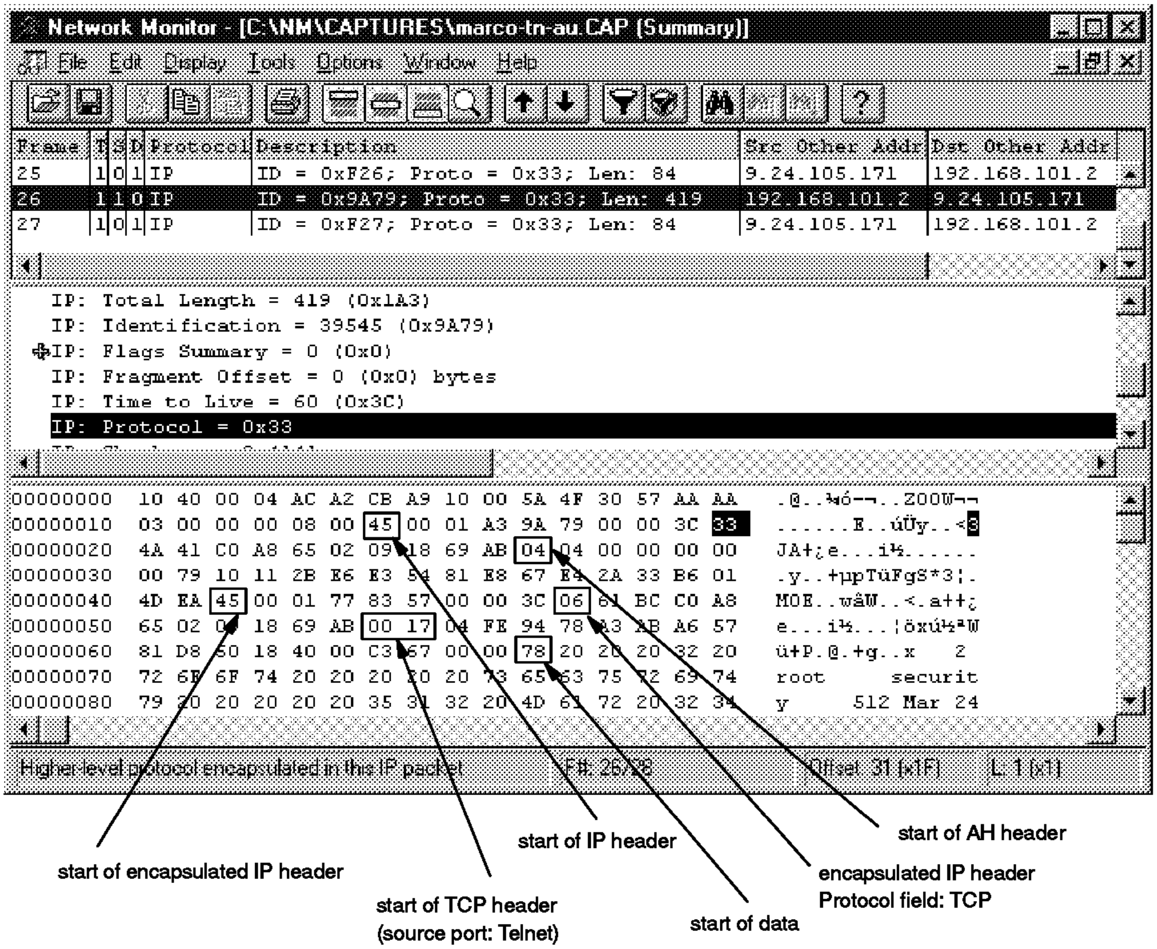
Network Monitor - [C:\NM\CAPTURES\marco-tn-plain.CAP (Detail)]
File Edit Display Tools Options Window Help

```
Frame T S D Protocol Description                             Src Other Addr  Dst Other Addr
34    1 1 0 TELNET   To Client With Port = 1278              192.168.101.2   9.24.105.171
35    1 0 1 TCP      .A...., len:    0, seq:2790752744-279   9.24.105.171    192.168.101.2
36    1 1 0 TELNET   To Client With Port = 1278              192.168.101.2   9.24.105.171

   IP: Total Length = 233 (0xE9)
   IP: Identification = 33853 (0x843D)
  +IP: Flags Summary = 0 (0x0)
   IP: Fragment Offset = 0 (0x0) bytes
   IP: Time to Live = 60 (0x3C)
   IP: Protocol = TCP - Transmission Control

00000000  10 40 00 04 AC A2 CB A9 10 00 5A 4F 30 57 AA AA   .@..¾ó-¬..ZOOW¬¬
00000010  03 00 00 00 08 00 45 00 00 E9 84 3D 00 00 3C 06   ......E..Tä=..<.
00000020  61 64 C0 A8 65 02 09 18 69 AB 00 17 04 FE 94 78   ad+¿e...i½...¦öx
00000030  B7 4D A6 57 81 E8 50 18 40 00 DB B9 00 00 20 20   +MªWüFP.@.¦¦..
00000040  20 31 20 72 6F 6F 74 20 20 20 20 20 73 79 73 74    1 root     syst
00000050  65 6D 20 20 20 20 20 20 20 20 32 31 20 4D 61 72   em         21 Mar
00000060  20 32 30 20 31 34 3A 34 30 20 75 6E 69 78 20 2D    20 14:40 unix -
00000070  3E 20 2F 75 73 72 2F 6C 69 62 2F 62 6F 6F 74 2F   > /usr/lib/boot/
00000080  75 6E 69 78 5F 75 70 0D 0A 64 72 77 78 72 2D 78   unix_up..drwxr-x

Higher-level protocol encapsulated in this IP packet    F#: 34/38    Off: 31 (x1F)    L: 1 (x1)
```

Figure 91. Trace of a Plain TCP/IP Connection

8.6.1.2 Example 2: AH Tunnel

We then activated an AH tunnel between the firewalls by setting the tunnel policy to auth only (see Chapter 4, “IBM eNetwork VPN Solutions” on page 59), and reopened the same telnet session. Network Monitor was no longer able to analyze anything beyond the outer IP header (see Figure 92 on page 168). However, it is obvious that AH is applied; the outer IP header carries protocol number 51 (0x33). Note that the payload is in cleartext.


Network Monitor - [C:\NM\CAPTURES\marco-tn-au.CAP (Summary)]
File Edit Display Tools Options Window Help

```
Frame T S D Protocol Description                             Src Other Addr  Dst Other Addr
25    1 0 1 IP       ID = 0xF26; Proto = 0x33; Len: 84       9.24.105.171    192.168.101.2
26    1 1 0 IP       ID = 0x9A79; Proto = 0x33; Len: 419     192.168.101.2   9.24.105.171
27    1 0 1 IP       ID = 0xF27; Proto = 0x33; Len: 84       9.24.105.171    192.168.101.2
```

```
  IP: Total Length = 419 (0x1A3)
  IP: Identification = 39545 (0x9A79)
 +IP: Flags Summary = 0 (0x0)
  IP: Fragment Offset = 0 (0x0) bytes
  IP: Time to Live = 60 (0x3C)
  IP: Protocol = 0x33
```

```
00000000   10 40 00 04 AC A2 CB A9 10 00 5A 4F 30 57 AA AA   .@..¾ó¬¬..ZOOW¬¬
00000010   03 00 00 00 08 00 45 00 01 A3 9A 79 00 00 3C 33   ......E..úÜy..<3
00000020   4A 41 C0 A8 65 02 09 18 69 AB 04 04 00 00 00 00   JA+¿e...i½......
00000030   00 79 10 11 2B E6 E3 54 81 E8 67 E4 2A 33 B6 01   .y..+µpTüFgS*3¦.
00000040   4D EA 45 00 01 77 83 57 00 00 3C 06 61 BC C0 A8   MOE..wâW..<.a++¿
00000050   65 02 04 18 69 AB 00 17 04 FE 94 78 A3 AB A6 57   e...i½...¦öxú½ªW
00000060   81 D8 50 18 40 00 C3 67 00 00 78 20 20 20 32 20   ü+P.@.+g..x   2
00000070   72 6F 6F 74 20 20 20 20 20 73 65 63 75 72 69 74   root     securit
00000080   79 20 20 20 20 20 35 31 32 20 4D 61 72 20 32 34   y     512 Mar 24
```

Figure 92. Trace of an AH Tunnel

It is not very difficult to identify the fields in the encapsulated IP packet. Have the packet format descriptions handy and consider these hints:

1. Network Monitor can highlight the outer IP header. The first byte usually has a hex value 0x45 (Version: 4, header length: five 32-bit words = 20 bytes). The 10th byte in the header indicates the protocol header that follows. This is AH (0x33) in our case.

2. Find the starting byte of the AH header. It is the first non-highlighted byte after the outer IP header (or, count a number of bytes indicated by the outer IP header length). This byte is the Next Header field; in our case IP (0x04), because AH is in tunnel mode.

3. The 2nd byte of the AH header shows its length. It is usually a 0x04 value, which means six 32-bit words = 24 bytes.

4. The four bytes starting at the 5th position contain the SPI (0x00000079). Following this things differ in function of the version of AH:

   - RFC 1826 compliant: ICV (in our case Keyed MD5, so it has 16 bytes)

   - New Draft compliant: 4 bytes of sequence number + ICV

5. After the AH header you find the encapsulated IP header. The same applies as for the outer IP header. Examining the 10th byte one can see that the next protocol header is TCP (0x06). Since the encapsulated IP header is also 20 bytes in length, the starting byte of the TCP header is quickly found. This,

together with the 2nd byte show the source TCP port. It is 0x0017 = 23, the well-known telnet port. Note that you can clearly see the source and destination IP addresses in the outer as well as in the inner IP header.

6. The upper nibble of the 13th byte in the TCP header indicates its length in 32-bit words. In our case this is 0x5, which means 20 bytes. With this information the start of the data can be located.

In summary, the packet structure is the following:

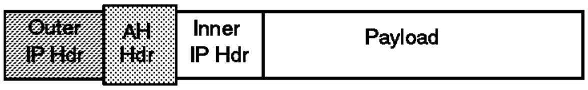
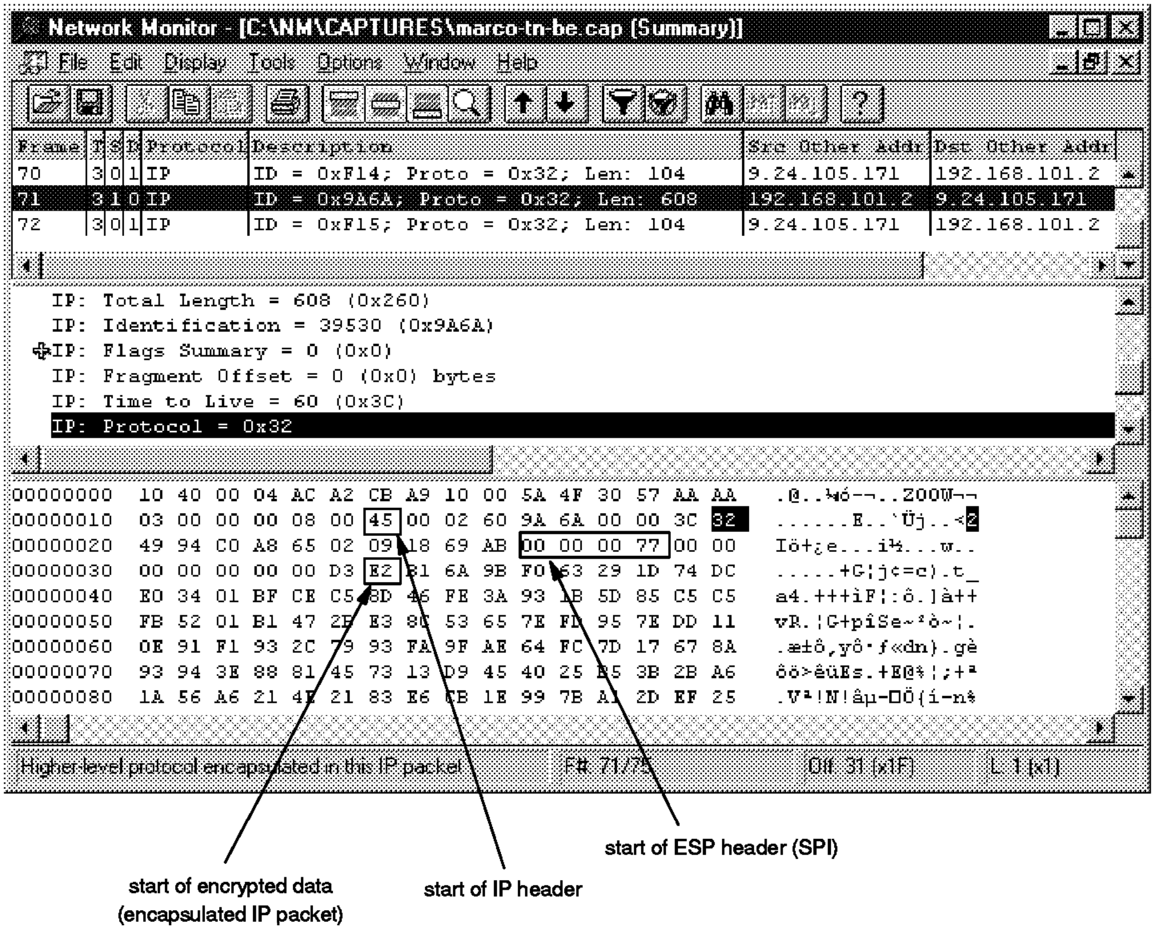


Figure 93. IP Packet Inside an AH Tunnel.

8.6.1.3 Example 3: ESP Tunnel

The same setup as in the AH tunnel example before, just the tunnel has been changed to ESP (encr only policy, see Chapter 4, “IBM eNetwork VPN Solutions” on page 59). Figure 94 shows a packet from the tunnel. Note that the inner IP packet is completely garbled. Nothing besides the destination can figure out the addresses, protocols or the data carried by it. The only identifiable fields are the 32-bit SPI and the 64-bit IV in the ESP header.


Network Monitor - [C:\NM\CAPTURES\marco-tn-be.cap (Summary)]
File
Edit
Display
Tools
Options
Window
Help

```
Frame T S I Protocol Description                             Src Other Addr  Dst Other Addr
70    3 0 1 IP       ID = 0xF14; Proto = 0x32; Len: 104      9.24.105.171    192.168.101.2
71    3 1 0 IP       ID = 0x9A6A; Proto = 0x32; Len: 608     192.168.101.2   9.24.105.171
72    3 0 1 IP       ID = 0xF15; Proto = 0x32; Len: 104      9.24.105.171    192.168.101.2
```

```
  IP: Total Length = 608 (0x260)
  IP: Identification = 39530 (0x9A6A)
 +IP: Flags Summary = 0 (0x0)
  IP: Fragment Offset = 0 (0x0) bytes
  IP: Time to Live = 60 (0x3C)
  IP: Protocol = 0x32
```

```
00000000  10 40 00 04 AC A2 CB A9 10 00 5A 4F 30 57 AA AA   .@..¼ó-¬..ZOOW¬¬
00000010  03 00 00 00 08 00 45 00 02 60 9A 6A 00 00 3C 32   .......E..`Üj..<2
00000020  49 94 C0 A8 65 02 09 18 69 AB 00 00 00 77 00 00   Iö+¿e...i½...w..
00000030  00 00 00 00 00 D3 E2 B1 6A 9B F0 63 29 1D 74 DC   .....+G¦j¢=c).t_
00000040  E0 34 01 BF CE C5 8D 46 FE 3A 93 1B 5D 85 C5 C5   a4.+++iF¦:ô.]à++
00000050  FB 52 01 B1 47 2B E3 8C 53 65 7E FD 95 7E DD 11   vR.¦G+pîSe~²ò~¦.
00000060  0E 91 F1 93 2C 79 93 FA 9F AE 64 FC 7D 17 67 8A   .æ±ô,yô·f«dn}.gè
00000070  93 94 3E 88 81 45 73 13 D9 45 40 25 B5 3B 2B A6   ôö>êüEs.+E@%¦;+ª
00000080  1A 56 A6 21 4E 21 83 E6 CB 1E 99 7B A1 2D EF 25   .Vª!N!âµ-ÐÖ{í-n%
```

Higher-level protocol encapsulated in this IP packet   F#: 71/75   Off: 31 (x1F)   L: 1 (x1)

start of ESP header (SPI)

start of encrypted data (encapsulated IP packet)

start of IP header

Figure 94. Trace of an ESP Tunnel

Note: When ESP is applied after AH (auth/encr policy, see Chapter 4, “IBM eNetwork VPN Solutions” on page 59), the trace looks exactly the same, because ESP in tunnel mode encrypts the whole original packet.

The outline of the packet is shown below. There is no ESP Authentication field, since the AIX Firewall does not yet support that feature.

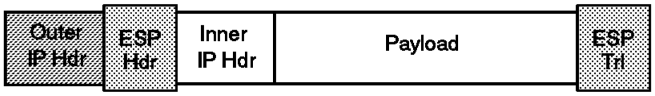
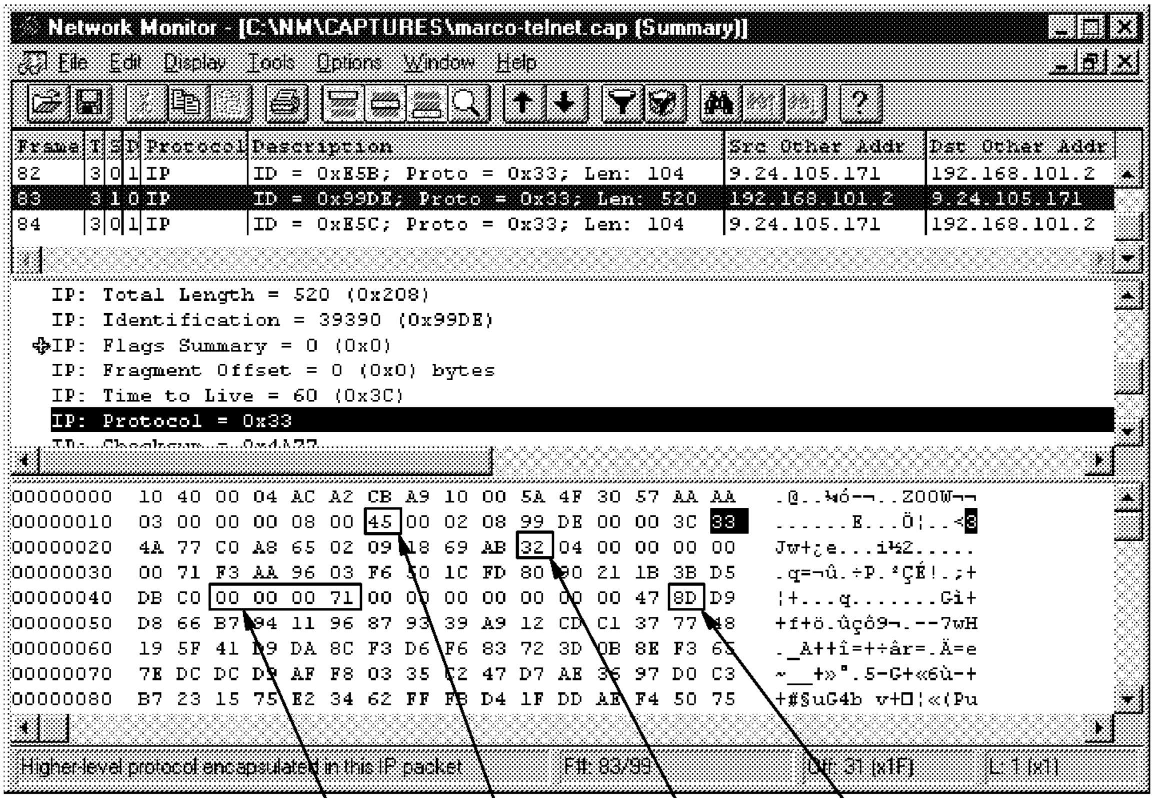


Figure 95. IP Packet Inside an ESP Tunnel

8.6.1.4 Example 4: Combined AH-ESP Tunnel

We modified the tunnel by setting the tunnel policy to encr/auth. The IPSec kernel first applies ESP and then AH in this case. Observe that the Next Header field in the AH header now indicates ESP (0x32). As in the previous case, we cannot examine the structure of the encapsulated IP packet.


Network Monitor - [C:\NM\CAPTURES\marco-telnet.cap (Summary)]
File
Edit
Display
Tools
Options
Window
Help

```
Frame T S D Protocol Description                              Src Other Addr   Dst Other Addr
82    3 0 1 IP       ID = 0xE5B; Proto = 0x33; Len: 104       9.24.105.171     192.168.101.2
83    3 1 0 IP       ID = 0x99DE; Proto = 0x33; Len: 520      192.168.101.2    9.24.105.171
84    3 0 1 IP       ID = 0xE5C; Proto = 0x33; Len: 104       9.24.105.171     192.168.101.2
```

```
  IP: Total Length = 520 (0x208)
  IP: Identification = 39390 (0x99DE)
 +IP: Flags Summary = 0 (0x0)
  IP: Fragment Offset = 0 (0x0) bytes
  IP: Time to Live = 60 (0x3C)
  IP: Protocol = 0x33
  IP: Checksum = 0x4A77
```

```
00000000  10 40 00 04 AC A2 CB A9 10 00 5A 4F 30 57 AA AA   .@..¾ó–¬..ZOOW¬¬
00000010  03 00 00 00 08 00 45 00 02 08 99 DE 00 00 3C 33   ......E...Ö¦..<3
00000020  4A 77 C0 A8 65 02 09 18 69 AB 32 04 00 00 00 00   Jw+¿e...i½2.....
00000030  00 71 F3 AA 96 03 F6 50 1C FD 80 90 21 1B 3B D5   .q=¬û.÷P.²ÇÉ!.;+
00000040  DB C0 00 00 00 71 00 00 00 00 00 00 00 47 8D D9   ¦+...q.......Gì+
00000050  D8 66 B7 94 11 96 87 93 39 A9 12 CD C1 37 77 48   +f+ö.ûçô9¬.--7wH
00000060  19 5F 41 D9 DA 8C F3 D6 F6 83 72 3D 0B 8E F3 65   ._A++î=+÷âr=.Ä=e
00000070  7E DC DC D9 AF F8 03 35 C2 47 D7 AE 36 97 D0 C3   ~__+»°.5-G+«6ù-+
00000080  B7 23 15 75 E2 34 62 FF FB D4 1F DD AE F4 50 75   +#§uG4b v+□¦«(Pu
```

Higher level protocol encapsulated in this IP packet   F#: 83/99   Off: 31 (x1F)   L: 1 (x1)

start of IP header

start of AH header (Next Protocol filed: ESP)

start of ESP header (SPI)

start of encrypted data (encapsulated IP packet)

Figure 96. Trace of a Combined AH-ESP Tunnel

Given the information above, we can now draw the outline of a packet in this tunnel:

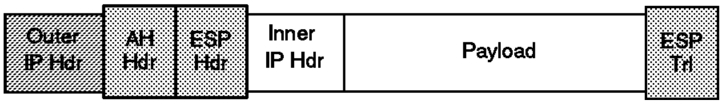
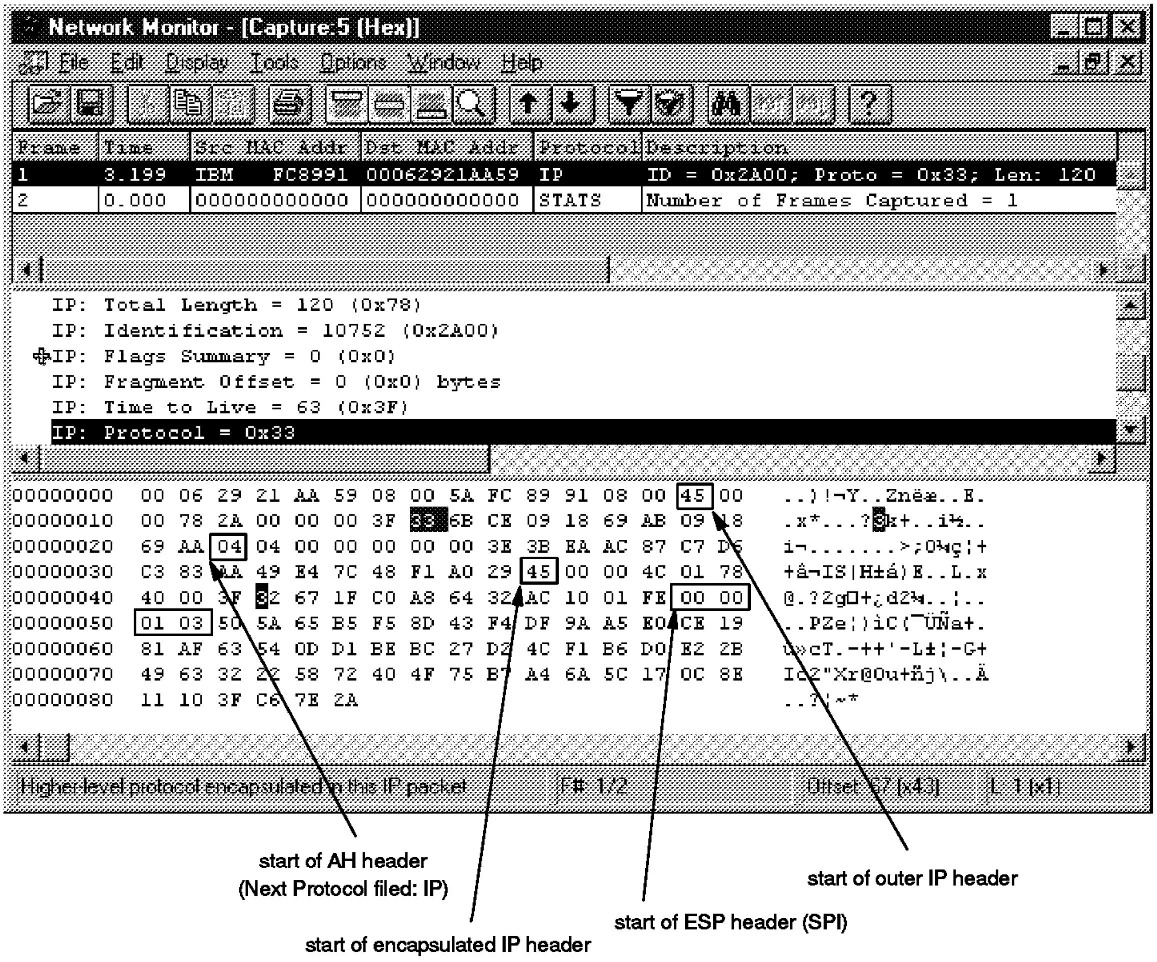


Figure 97. IP Packet Inside a Combined AH-ESP Tunnel

8.6.1.5 Example 5: Nested ESP Packet Inside AH Tunnel

This example shows a trace of the environment as described in 6.2, “Scenario Setup” on page 116 for the business partner/supplier network. The original IP datagram is first encrypted at the source host using ESP in transport mode. That datagram is then sent to the firewall where AH in tunnel mode is applied to it, thus adding an outer IP header with IP address information of the firewalls.

In contrast to the AH tunnel shown in 8.6.1.2, “Example 2: AH Tunnel” on page 167, the Next Header field of the inner IP header now points to ESP instead of TCP.


Network Monitor - [Capture:5 (Hex)]
File Edit Display Tools Options Window Help
start of AH header
(Next Protocol filed: IP)
start of encapsulated IP header
start of ESP header (SPI)
start of outer IP header

```
Frame  Time   Src MAC Addr  Dst MAC Addr  Protocol Description
1      3.199  IBM   FC8991  00062921AA59  IP       ID = 0x2A00; Proto = 0x33; Len: 120
2      0.000  000000000000  000000000000  STATS    Number of Frames Captured = 1
```

```
  IP: Total Length = 120 (0x78)
  IP: Identification = 10752 (0x2A00)
 +IP: Flags Summary = 0 (0x0)
  IP: Fragment Offset = 0 (0x0) bytes
  IP: Time to Live = 63 (0x3F)
  IP: Protocol = 0x33
```

```
00000000   00 06 29 21 AA 59 08 00 5A FC 89 91 08 00 45 00    ..)!¬Y..Znëæ..E.
00000010   00 78 2A 00 00 00 3F 33 6B CE 09 18 69 AB 09 18    .x*...?3k+..i½..
00000020   69 AA 04 04 00 00 00 00 00 3E 3B EA AC 87 C7 D6    i¬.......>;O¼ç¦+
00000030   C3 83 AA 49 E4 7C 48 F1 A0 29 45 00 00 4C 01 78    +â¬IS|H±á)E..L.x
00000040   40 00 3F 32 67 1F C0 A8 64 32 AC 10 01 FE 00 00    @.?2g□+¿d2¼..¦..
00000050   01 03 50 5A 65 B5 F5 8D 43 F4 DF 9A A5 E0 CE 19    ..PZe¦)iC(¯ÜÑa+.
00000060   81 AF 63 54 0D D1 BE BC 27 D2 4C F1 B6 D0 E2 2B    ü»cT.-++'-L±¦-G+
00000070   49 63 32 22 58 72 40 4F 75 B7 A4 6A 5C 17 0C 8E    IcZ"Xr@Ou+ñj\..Ä
00000080   11 10 3F C6 7E 2A                                  ..?¦~*
```

```
Higher level protocol encapsulated in this IP packet   F#: 1/2   Offset: 67 (x43)   L: 1 (x1)
```

Figure 98. Trace of a Nested ESP Packet Inside an AH Tunnel

Given the information above, we can now draw the outline of a packet in this nested tunnel:

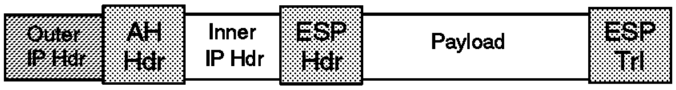


Figure 99. ESP Packet Nested Inside an AH Tunnel

8.7 Features That May Not Work with IPSec

There are some features present in most available TCP/IP implementations that are useful for various purposes but may not work in conjunction with IPSec. The following sections list some of them and describe the problems that can be encountered in an IPSec environment.

8.7.1 Path MTU Discovery and IP Fragmentation

When TCP calculates the maximum segment size (MSS) it doesn′t know about the extra length needed for IPSec headers. MTU path discovery (described in RFC 1191) turns on the ′Do not fragement′ bit in the IP header in the TCP layer. Then the IP layer calls the IPSec kernel and adds the IPSec headers. If this added length exceeds an MTU, it will be rejected because of the ′Do not fragment′ bit set. So TCP will retry after reducing the MSS, but it uses the MTU returned in the ICMP response and does not take into account the IPSec header length, so the retry won′t work.

When IP datagrams carrying IPSec traffic are fragmented along the way, the general rule is to apply IPSec before fragmentation for outbound datagrams, and to apply IPSec after reassembly for inbound datagrams. This must be done in order to assure proper IPSec processing for authentication and/or encryption/decryption.

See the current Internet Drafts for IPSec, AH and ESP on the issues of fragmentation and path MTU discovery.

8.7.2 Traceroute

According to the current IPSec specifications, there are certain ICMP message types that can be accepted over IPSec connections according to the local policy. You need to check with the vendor of your specific IPSec implementation if they support traceroute over IPSec.

Note: This obviously applies only to IPSec traffic in transport mode. Tunneled IPSec packets that are just passed through from one interface to another in a security gateway may carry traceroute traffic undisturbed. However, the whole purpose of traceroute is lost within IPSec tunnels because only the first gateway and the final destination, but no other intermediate gateways will respond to those packets.

8.7.3 Network Address Translation

We have described some issues with NAT and IPSec in 1.3.3.1, “Network Address Translation” on page 14. The following reference provides additional information on NAT and its implications when used in conjunction with IPSec:

```
http://www.ietf.org/internet-drafts/draft-moskowitz-net66-vpn-00.txt
```

Chapter 9. Additional VPN Configurations with IBM Products

In this chapter, we describe briefly how to set up IPSec connections between IBM eNetwork VPN solutions that we have not used throughout the previous scenarios. Please refer to Table 3 on page 76 for a complete overview of possible combinations. Also note that any VPN scenarios involving the 2210/2216 Router will be described in a separate redbook, *A Comprehensive Guide to Virtual Private Networks, Volume II: IBM Nways Router Solutions*, SG24-5234, to be published later this year.

9.1 Additional eNetwork Firewall for AIX Combinations

This section describes possible VPN scenarios involving the eNetwork Firewall for AIX that have not been previously discussed.

9.1.1 Export File Format

The names of the export files for a manual tunnel on a firewall V3.1 are:

- fwexpmctx.manual
- fwexppolicy
- fwexppolicy.3.1

The format of the fwexpmctx.manual file is identical to the format used on the OS/390 Server and is described in 9.3.5, "FWTUNNL Export File Formats" on page 177. The third file fwexppolicy.3.1 is equivalent to fwexppolicy on the OS/390 Server and the second file fwexppolicy is very similar to the third file fwexppolicy.3.1; it just has no timeout field.

Note: The format of the export files on AIX V4.3 is slightly different because of additional information necessary for the new IPSec header formats, replay prevention and ESP authentication option.

9.1.2 Interoperability between eNetwork Firewall for AIX and AIX V4.3

The best way to connect the two products is an IBM tunnel. Build the network objects, tunnel definition and packet filters on the eNetwork Firewall for AIX as described in Chapter 5, "Branch Office Connection Scenario" on page 81. Change the filter rules on the firewall as described in 5.7.1, "Allow Only Firewall-to-Firewall Traffic" on page 108. This is necessary, because the filter rules need to allow the firewall itself to have traffic to the AIX V4.3 system. Export the definition and transfer it to the AIX V4.3 system.

On the AIX V4.3 server you need to perform the following steps. As the export file formats between AIX V4.3 and the firewall differ, you need to issue the import command with the -n option on the AIX V4.3 system as follows: `imptun -f /imptun -n`. In this case /imptun is the directory in which you placed the export files after transferring them to the AIX V4.3 system. The filter rules for all tunnel-related traffic will be automatically generated when the tunnel is defined via the import command. For all other activities see 6.4.1, "Configuration of the AIX V4.3 Server" on page 120.

All possible values that you may choose during the IBM tunnel definition on the firewall are supported by AIX V4.3 as well, therefore you may just use values that best meet your needs.

Our test was conducted with the policy encr/auth and the ESP transform CDMF. The firewall will always use the AH transform keyed MD5.

After activating the tunnels on both sides try a ping between the two systems to see whether the tunnel works. Another possibility is the admin_test command on the firewall.

9.2 Additional AIX V4.3 Combinations

This section describes possible VPN scenarios involving the AIX V4.3 that have not been previously discussed.

9.2.1 Interoperability between Two AIX V4.3 Systems

For a connection between two AIX V4.3 systems in our opinion there are two alternatives:

1. For easy administration choose IBM Tunnel and DES CBC 8.
2. For even stronger algorithms choose Manual Tunnel, HMAC SHA and 3DES.

The procedure to set up a tunnel is easy. On the first system follow the steps described in 6.4.1, “Configuration of the AIX V4.3 Server” on page 120. Export the tunnel definition and transport the file to the second AIX system. Follow the above procedure again on the second system, but import the tunnel definition instead of defining a new one. All necessary packet filters will be automatically created and activated by the system.

9.3 OS/390 Server Combinations

When it comes to manual tunnels the OS/390 Server has the same capabilities as the eNetwork Firewall for AIX. The syntax and behavior of the commands used in the following setup are exactly the same as on the eNetwork Firewall for AIX. Before trying to implement the setup described in the next section we assume that you have read and understood Chapter 5, “Branch Office Connection Scenario” on page 81.

9.3.1 Interoperability with the eNetwork Firewall for AIX

The configuration described in this section allows tunnel traffic between the OS/390 server and the eNetwork Firewall for AIX. Please follow the steps described in 9.3.1.1, “OS/390 Server IPSec Configuration” on page 175 to establish the tunnel.

Start on the AIX system. Create the network objects, tunnel definition and packet filters as shown in Chapter 5, “Branch Office Connection Scenario” on page 81 (take FW1 as an example). We used the following parameters:

- IP address AIX: 9.24.105.171
- IP address OS/390: 9.24.105.77
- Tunnel type: Manual
- Policy: Encr/auth

- ESP transform: CDMF

Note: Because the manual tunnel features on both systems are currently identical (RFCs 18xx, same policies and transforms), every combination will work as long as you use a manual tunnel.

Change the filter rules on the firewall as described in 5.7.1, "Allow Only Firewall-to-Firewall Traffic" on page 108. This is necessary, because the filter rules we describe in this example only allow firewall-to-firewall traffic.

Export the tunnel definition and transfer it to the OS/390 server.

Note: You need the following modification on the OS/390 Server for the import to work. Copy the fwexppolicy.3.1 file to fwexppolicy. The OS/390 Server does not use the fwexppolicy.3.1 file, but expects its contents in the fwexppolicy file. Otherwise your import will fail.

9.3.1.1 OS/390 Server IPSec Configuration

The steps on the OS/390 server are:

1. If not already done, define the secure adapter to the system (in our case IP address 192.168.236.29):

```
fwadapter cmd=change addr=192.168.236.29 state=secure
```

2. Create a network object for the non-secure interface of the local S/390 system:

```
fwnwobj cmd=add name=mvs28c desc="S/390 Server"
        type=firewall addr=9.24.105.77 mask=255.255.255.255
```

3. Define the network object for the AIX firewall:

```
fwnwobj cmd=add name=rs600024e desc="AIX Firewall"
        type=firewall addr=9.24.105.171 mask=255.255.255.255
```

4. Import the tunnel definition. (In our case, the tunnel definition was transferred to /u/karl and the tunnel ID was 390.)

```
fwtunnl cmd=import directory=/u/karl tunnel=390
```

5. Now you need to create the packet filter rules:

```
fwfrule cmd=add name=ESPtraffic desc="ESP traffic"
        type=permit protocol=ESP srcopcode=any srcport=0 destopcode=any
        destport=0 interface=nonsecure routing=local direction=both
```

```
fwfrule cmd=add name=AHtraffic desc="AH traffic"
        type=permit protocol=ah srcopcode=any srcport=0 destopcode=any
        destport=0 interface=nonsecure routing=local direction=both
```

```
fwfrule cmd=add name=tunneltraffic desc="permit all thru tunnel 390"
        type=permit protocol=all srcopcode=any srcport=0 destopcode=any
        destport=0 interface=nonsecure routing=local direction=both
        tunnel=390
```

6. In order to be able to build the service we need the IDs assigned for the above rules:

```
fwfrule cmd=list
```

   In our case we got 502, 503 and 501.

7. The next step combines the filters rules you just added into a service. This service describes all of the tunnel traffic that is permitted into and out of the local system.

```
fwservice cmd=create name=alltunneltraffic desc="All tunnel traffic"
          rulelist=502/f,502/b,503/f,503/b,501/f,501/b
```

Note: The order of the filter rules in the rule list is important. Filter rules associated with the IPSec protocols are always placed at the beginning of the list. In our example, these are rules 502 and 503.

8. Use the fwservice command to obtain the ID assigned to the service you just created:

   ```
   fwservice cmd=list name=alltunneltraffic
   ```

   In our case the assigned ID is 501.

9. Find out the assigned IDs for the defined network objects and filter rules:

   ```
   fwnwobj cmd=list
   ```

   In our case, the network objects got the numbers 501 and 502.

10. The fwconns command is used to create a connection to associate the local and remote host objects with the filter rule service:

    ```
    fwconns cmd=create name=alltunneltraffic
            desc="All tunnel traffic (tunnel=390)"
            source=mvs28c destination=rs600024e servicelist=501
    ```

11. The tunnel setup is now completed and the tunnel needs to be activated:

    ```
    fwtunnl cmd=activate tunnel=390
    ```

12. Finally, use the fwfilter command to activate the filter rules that you created:

    ```
    fwfilter cmd=update
    ```

9.3.2 Interoperability with AIX V4.3

Try the following values:

- Tunnel type: Manual
- Policy: Encr/auth (every policy should work)
- AH transform: Keyed MD5
- ESP transform: DES CBC 8, DES CBC4 or CDMF
- New header format: No (will be assigned automatically because of the AH transform)
- Replay prevention: No (will be also assigned automatically)

This combination has been tested successfully by IBM development.

Note: When you import a tunnel definition from OS/390 Server to AIX V4.3, you must first remove the tunnel life time value from the fwexppolicy file, then use the following command: `imptun -n`.

9.3.3 Interoperability with the eNetwork Communications Suite

Try the following values:

- Tunnel type: Manual
- Policy: Encr/auth, encr only or auth only
- AH transform: Keyed MD5
- ESP transform: DES CBC 8 or DES CBC4

This combination has been tested successfully by us and also by IBM development.

To make it easier to identify matching configuration values between the two systems, take a look at Table 6. Note that SPI values are decimal on OS/390 but hexadecimal on Comms Suite.

Table 6. Matching Configuration Items between OS/390 Server and eNetwork Communications Suite

OS/390 Server	**eNetwork Communications Suite**
owner IP	remote IP
partner IP	local IP
owner SPI	remote SPI
partner SPI	local SPI
owner key	remote key
partner key	local key
owner transform	remote transform
partner transform	local transform

9.3.4 Interoperability with Other Vendors′ Products

If you find matching values for IPSec features such as header format, transform and policy, you should first set up the tunnel definition on the OS/390 server and then export the definition. Examine the export files and match their values in the partner tunnel definition. See the next section for an explanation of the export files.

You can find additional information on how to establish IPSec connections from your OS/390 Server to a non-IBM system at the following Web page:

`http://www.s390.ibm.com/products/mvs/firewall/ipsec.htm`

9.3.5 FWTUNNL Export File Formats

The fwtunnl command is used to export one or more tunnel definitions. The export action creates two files in the requested directory.

Note: The format of the export files may change with future versions of IPSec-enabled IBM platforms, so please be careful with the descriptions shown in this section.

The first file is known as the context file and is named fwexpmctx.manual. This file contains 22 lines (per tunnel definition) and all data must begin in column 1. Unless stated otherwise all values are in decimal. An example of the format and typical contents of the fwexpmctx.manual file follows:

```
Typical Contents:                   File Format:
#Tunnel_partner Version = 1.0   Line 1: context line - must be: #Tunnel_partner Version = 1.0
11.130.10.7                          2: IP Address of Tunnel Owner
11.130.10.5                          3: IP Address of Tunnel Partner
1                                    4: Tunnel Id
682262                               5: ESP SPI of Tunnel Owner
682262                               6: AH SPI of Tunnel Owner
1000                                 7: ESP SPI of Tunnel Partner
1000                                 8: AH SPI of Tunnel Partner
CDMF                                 9: Encryption Algorithm of Tunnel Owner (CDMF or DES_CBC_8 or DES_CBC_4)
8                                   10: Size (number of bytes) of Encryption Key of Tunnel Owner
0xce803234ce732f25                  11: Encryption Key of Tunnel Owner (hex value starting with '0x')
CDMF                                12: Encryption Algorithm of Tunnel Partner (CDMF or DES_CBC_8 or DES_CBC_4)
8                                   13: Size (number of bytes) of Encryption Key of Tunnel Partner
0xf2e943e6c8c24fa7                  14: Encryption Key of Tunnel Partner (hex value starting with '0x')
KEYED_MD5                            15: Authentication Algorithm of Tunnel Owner (KEYED_MD5)
16                                  16: Size (number of bytes) of Authentication Key of Tunnel Owner
0x3ba30d2a8278c94b29828ccef98081af  17: Authentication Key of Tunnel Owner (hex value starting with '0x')
KEYED_MD5                            18: Authentication Algorithm of Tunnel Partner (KEYED_MD5)
16                                  19: Size (number of bytes) of Authentication Key of Tunnel Partner
0xc1bbc0b0105bf88387f8708ca2585897  20: Authentication Key of Tunnel Partner (hex value starting with '0x')
0                                   21: Reserved - set to 0
28800                               22: Timeout in Seconds
```

Figure 100. Example of the Export File fwexpmctx.manual

Note: It is possible to export multiple tunnel definitions at one time. In this case, the subsequent tunnel policy definitions are simply appended to the bottom of the file.

The second file created by the export request is known as the policy file and is named fwexppolicy. This file contains one line (per tunnel definition) and all data must begin in column 1. Unless stated otherwise all values are in decimal. Each line contains eight fields, each separated by a blank. An example of the format and typical contents of the fwexppolicy file follows:

Typical Contents in Fields 1 through 8

The content of a typical fwexppolicy is:

```
                              Fields
--------------------------------------------------------------------
     1               2        3     4       5     6    7       8
--------------------------------------------------------------------
 11.130.10.5     11.130.10.7  1   28800    n/y   n/n   y   key=manual
```

File Format of Fields 1 through 8:

```
 Field 1 - IP Address of Tunnel Partner
 Field 2 - IP Address of Tunnel Owner
 Field 3 - Tunnel ID
 Field 4 - Timeout in Seconds
 Field 5 - Encryption/Authentication Flag - valid combinations are:
           y/y - Encryption and Authentication
           y/n - Encryption Only
           n/y - Authentication Only
 Field 6 - Encryption Mode/Authentication Mode must be n/n (tunnel mode)
 Field 7 - Authentication First - y or n
           Determines the order of operations when
           both Encryption and Authentication are requested.
 Field 8 - Tunnel Type - must be: key=manual
```

Figure 101. Example of an Export fwexppolicy File

Note: It is possible to export multiple tunnel definitions at one time. In this case, the subsequent tunnel definitions are appended to the bottom of the file.

For further information on OS/390 Server see the redbook *Stay Cool on OS/390: Installing Firewall Technology*, SG24-2046.

9.4 Additional OS/2 TCP/IP V4.1 IPSec Client Combinations

In this section, we describe previously undocumented features of the OS/2 TCP/IP V4.1 IPSec Client. The IPSec code in the OS/2 TCP/IP V4.1 IPSec Client actually provides more functionality than being only a dynamic tunnel client to an eNetwork Firewall for AIX. It can also be configured for manual IPSec tunnels. We provide the necessary information to configure such manual tunnels which will then allow the following IPSec combinations:

1. Tunnel between OS/2 TCP/IP V4.1 IPSec Client and eNetwork Firewall for AIX
2. Tunnel between OS/2 TCP/IP V4.1 IPSec Client and OS/390 Server
3. Tunnel between OS/2 TCP/IP V4.1 IPSec Client and AIX V4.3
4. Tunnel between two OS/2 TCP/IP V4.1 IPSec Clients
5. Tunnel between OS/2 TCP/IP V4.1 IPSec Client and eNetwork Communications Suite

Finally, the packet filtering capability of the OS/2 TCP/IP V4.1 IPSec Client can be used to turn OS/2 into a mini-firewall. We first describe the manual tunnel setup, then the possible combinations, and at the end of this section the packet filtering capabilities of OS/2 TCP/IP V4.1.

Note: When operating the OS/2 TCP/IP V4.1 IPSec Client with manual IPSec tunnels, do not use the Secure Remote Client Configuration program because that can only be used to connect to an eNetwork Firewall for AIX via SSL and obtain a dynamic tunnel configuration.

9.4.1 Configuring IPSec Filters and Tunnels

IPSec on OS/2 TCP/IP 4.1 consists of four device drivers and associated configuration utilities. Two of the device drivers are required and the other two are implementations of encryption and authentication algorithms (CDMF, MD5) and are therefore optional. The two required device drivers for IPSec are a filter device driver, FWIP.SYS, and a device driver that provides the framework for IPSec, IPSEC.SYS. To manually configure IPSec tunnels on OS/2 three components must be configured:

1. Filters to filter IP packets through a tunnel
2. Tunnel policy to define tunnel endpoint IP addresses and whether to apply encryption and/or authentication and in what order, and tunnel modes
3. Tunnel context containing keys, tunnel ID and algorithms to be applied

The format of the filters and tunnel definition files is very similar to that on the eNetwork Firewall for AIX and OS/390 Server. We have included it in this section again for your convenience and for the sake of completeness.

> **Tip**
>
> If you have a system running eNetwork Firewall for AIX, configure the OS/2 system as a remote IPSec client to the firewall (see 7.3.3.5, "Defining the IPSec Connection to the Firewall" on page 144) and activate an IPSec connection (see 7.4, "Activating and Deactivating the Dynamic Tunnel" on page 145). This will transfer a tunnel policy and tunnel context file to the OS/2 system and create the necessary IPSec filters file. Save those files before you close the tunnel (whereupon they will be deleted) and use them as a base to configure your manual IPSec connections.

9.4.1.1 Setting Up IP Filtering

The filter device driver, FWIP.SYS, when active, is called by the IP layer for each IP packet coming into the workstation and going out of the workstation. The filter device driver will determine, based on configurable rules, whether an IP packet is allowed to continue. The packet is compared to the rules starting with the first and then each subsequent rule until a match is found. When an exact match is found, the action (permit or deny) is performed. If no rule matches the packet is denied by default.

Note: Filter rules are described fully in *eNetwork Firewall for AIX, Reference Guide*, SC31-8418. What is described here will be what is necessary to filter IP packets through an IPSec tunnel.

The only facility that exists today to build filter rules on OS/2 is an editor. Once the filters file is built, the executable CFGFILT.EXE can be called to process the file and deliver the configuration to the filter device driver. CFGFILT.EXE takes the following parameters:

-u To update the filter rules in the device driver using contents of mptnetcsecurityfwfiltrs.cnf.

-i To initialize the filter device driver; must be used with -u.

-f [**file**] To check a set of rules. Default file is mptnetcsecurityfwfiltrs.cnf.

-c To deactivate active filter rules and go to default rules of deny everything.

-m [#] Maximum concurrent Real Audio connections.

-p Real Audio port.

-d [{**start** | **stop**}] Start or stop filter logging; default, -d with no option, is to start.

<no parameter> Default action is to dump active filter rules (no options)

The filter rules to support a local host (1.2.3.4) to remote host (4.3.2.1) VPN (tunnel 10) are the following (lines have been indented to fit the page):

```
# This is a comment as are all lines that start with #
# These are the rules for the gateway with non-secure address 001.002.003.004

# ip packets that are ipsec esp protocol from/to a known tunnel end on the
  non-secure interface
permit 1.2.3.4 255.255.255.255 4.3.2.1 255.255.255.255 esp any 0 any 0
  non-secure local outbound
permit 4.3.2.1 255.255.255.255 1.2.3.4 255.255.255.255 esp any 0 any 0
  non-secure local inbound

# ip packets that are ipsec ah protocol from/to a known tunnel end on the
  non-secure interface
permit 1.2.3.4 255.255.255.255 4.3.2.1 255.255.255.255 ah any 0 any 0
  non-secure local outbound
permit 4.3.2.1 255.255.255.255 1.2.3.4 255.255.255.255 ah any 0 any 0
  non-secure local inbound

# ip packets that are from/to non-secure interface to/from known tunnel
  end through tunnel 10
permit 1.2.3.4 255.255.255.255 4.3.2.1 255.255.255.255 all any 0 any 0
  non-secure local both t=10
permit 4.3.2.1 255.255.255.255 1.2.3.4 255.255.255.255 all any 0 any 0
  non-secure local both t=10

# default final rule is deny everything.
```

Figure 102. OS/2 IP Filter Rules

The format of filter rules on OS/2 resembles that of the eNetwork Firewall for AIX. You can take a look at the /etc/security/fwfilters.cnf on a firewall to get a better idea of how to configure filter rules on OS/2.

The rules for the other end of the tunnel would be the same with reversal of source and destination addresses and masks.

All interfaces (IP addresses) on the system are assumed to be non-secure. To define a secure interface, please see 9.4.7, “Turning OS/2 Into a Mini-Firewall” on page 188.

To define and activate filters on a tunnel endpoint do the following:

1. Edit the file mptnetcfwfiltrs.cnf adding the above lines with correct IP addresses and masks.
2. Enter on a command line: `cfgfilt -i -u -d` to activate filters, load the filter rules and activate logging.

There is a switch available in the IP layer to control filters. To have the IP layer invoke filters when FWIP.SYS is loaded, enter the following on the command line: `inetcfg -s firewall 1`.

To prevent the IP layer from invoking filters enter the following on the command line: `inetcfg -s firewall 0`.

To check the current setting of this switch enter the following on the command line: `inetcfg -g firewall`.

CFGFILT, when loading filter rules issues inetcfg -s firewall 1.

9.4.1.2 Setting Up IPSec Tunnel Policies

The tunnel policy is implemented in FWIP.SYS along with filters and consists of settings that control how IPSEC.SYS processes outbound IP packets that are filtered through a tunnel and indicates the level of protection required to FWIP.SYS for inbound IP packets that are filtered through a tunnel.

The policy file is created with an editor and has the following format (line has been indented to fit the page):

```
<Source IP Address> <Target IP Address> <Context ID> <Encrypt/Auth>
   <Encrypt Mode/Auth Mode> <MAC first>
```

Source IP address The IP address, either in qaud format or alias name, of the source host associated with the policy entry (this host IP address).

Target IP address The IP address, either in qaud format or alias name, of the target host associated with the policy entry (other end of the tunnel host IP address).

Context ID The tunnel context ID which identifies tunnel ID in the policy file.

Encrypt A y or Y in this field indicates encrypted data. The other option available is n or N.

Auth A y or Y in this field indicates authenticated data. The other option available is n or N.

Encrypt Mode A y or Y in this field indicates transport mode and an n or N indicates tunnel mode for encrypted packet.

Auth Mode A y or Y in this field indicates transport mode and an n or N indicates tunnel mode for authenticated packet.

MAC first A y or Y in this field indicates authentication is done before encryption. An n or N indicates encryption is done first.

For example:

```
   1.2.3.4  4.3.2.1  10  y/y  n/n y
```

The above example would be the policy for tunnel endpoint 1.2.3.4. It indicates the tunnel endpoints as 1.2.3.4 and 4.3.2.1, the tunnel ID 10, to perform encryption and authentication, use tunnel mode (IP packet in an IP packet) and perform authentication before encryption, that is, authenticate clear text.

The format of tunnel policy files on OS/2 resembles that of the eNetwork Firewall for AIX and OS/390 Server, but without the timeout value and key=manual statement. On AIX V4.3, there is no separate policy file so you have to create one if you want to establish an IPSec connection to such a system.

Each additional policy statement would go on a new line in the same file.

To enter the tunnel policy into the FWIP.SYS device driver use the utility FWINSERT.EXE. The only parameter FWINSERT.EXE takes is the file name of the file containing the policy.

To create and load tunnel policy do the following:

1. Edit the file mptnetcsecuritypolicy adding the desired policy for each tunnel supported on a new line.
2. Enter on a command line `fwinsert mptnetcsecuritypolicy`.

Each time FWINSERT is called it completely replaces the existing policy entries with the contents of the file given.

If a tunnel policy indicates that transport mode is to be used, MTU path discovery must be turned off (see 8.7, "Features That May Not Work with IPSec" on page 172). To do this enter the following on the command line: `inetcfg -s mtudiscovery 0`.

To turn mtudiscovery back on enter the following on the command line: `inetcfg -s mtudiscovery 1`.

9.4.1.3 Setting Up the IPSec Tunnel Context Cache

Tunnel context cache is implemented in IPSEC.SYS along with the framework for IPSec. A tunnel context entry defines all the parameters needed by IPSEC.SYS to encrypt, decrypt and authenticate an IPSec IP packet.

To create a tunnel context entry a file must be edited to contain the following fields, each on a new line. Any line beginning with a # is treated as a comment.

```
Line  1 : The IP address, in dotted format or alias name, of the other end of
          the tunnel.
Line  2 : The IP address, in dotted format or alias name, of this host.
Line  3 : Tunnel ID.
Line  4 : This host's Security Parameter Index for ESP (Encryption).
Line  5 : This host's Security Parameter Index for AH ( Authentication).
Line  6 : The other end of the tunnel host's Security Parameter Index for ESP
          (Encryption).
Line  7 : The other end of the tunnel host's Security Parameter Index for AH
          (Authentication).
Line  8 : This host's encryption algorithm.
          Supported in Internet distributed version of TCP/IP:           CDMF
Line  9 : This host's encryption key length.
          Must be: 8
Line 10 : This host's encryption key. (hexadecimal integer)
          Must be hex characters so will be 16 characters in length.
Line 11 : The other end of the tunnel host's encryption algorithm.
          Supported in Internet distributed version of TCP/IP:           CDMF
Line 12 : The other end of the tunnel host's encryption key length.
          Must be: 8
Line 13 : The other end of the tunnel host's encryption key.
          Must be hex characters so will be 16 characters in length.
Line 14 : This host's Authentication (mac) algorithm.
          Supported: KEYED_MD5
Line 15 : This host's Authentication (mac) key length.
          Must be: 16
Line 16 : This host's Authentication (mac) key.
          Must be hex characters so will be 32 characters in length.
Line 17 : The other end of the tunnel host's Authentication (mac) algorithm.
          Supported: KEYED_MD5
Line 18 : The other end of the tunnel host's Authentication (mac) key length.
          Must be: 16
Line 19 : The other end of the tunnel host's Authentication (mac) key.
```

```
          Must be hex characters so will be 32 characters in length.
Line 20 : Start (Time).
          If zero, use current time.  Represented as number of seconds since
          Jan 1, 1970.  This value is used with the value on Line 21 to
          determine the tunnel end time.  The tunnel context for this record
          will be active as soon as hand_k is run.
Line 21 : End (Time).
          If start time is zero, it is key life time in seconds.  Represented
          as number of seconds since Jan1, 1970.  Recommended key lifetime is
          8 hours (28800 seconds) or less.
Line 22 : Reserved.
          Must be: 0.0.0.0
```

For example:

```
# Tunnel Context for tunnel 10
4.3.2.1                            # 1.  The other tunnel endpoint’s IP Address
1.2.3.4                            # 2.  This host’s IP Address
10                                 # 3.  Tunnel ID
400                                # 4.  This host’s Security Parameter Index (Encr)
400                                # 5.  This host’s Security Parameter Index  (Auth)
668829                             # 6.  Other host’s Security Parameter Index (Encr)
668829                             # 7.  Other host’s Security Parameter Index (Auth)
CDMF                               # 8.  This host’s  Encryption Algorithm
8                                  # 9.  This host’s  Encryption Key Length
0x000079140006a0ac                 # 10. This host’s Encryption Key
CDMF                               # 11. The other host’s Encryption Algorithm
8                                  # 12. The other host’s Encryption Key Length
0x1234da442443212c                 # 13. The other host’s Encryption Key
KEYED_MD5                          # 14. This host’s  Authentication (MAC) Algorithm
16                                 # 15. This host’s MAC Length
0x618a751f4411818d41ce388ed7bfc9fd # 16. This host’s MAC Key
KEYED_MD5                          # 17. The other host’s MAC Algorithm
16                                 # 18. The other host’s MAC Length
0xa8e3377a51605476683fb8d269c21023 # 19. The other host’s MAC Key
0                                  # 20. Start - Current Time
28800                              # 21. End - 8 hours from now
0.0.0.0                            # 22. Reserved - Must be 0.0.0.0
```

The format of tunnel context files on OS/2 resembles that of the eNetwork Firewall for AIX and OS/390 Server, but line 22 must be added manually. Modify an AIX V4.3 tunnel export file according to the format shown in the example above.

There are a few utilities to manage the contents of the tunnel context cache.

The HAND_K.EXE Command:

HAND_K.EXE is used to enter entries into the tunnel context cache. It takes as input a file containing one or more tunnel context entries. This interface was meant to be a programmed interface to a GUI and does not produce error messages that are particularly helpful. Successful processing of the file will result in no messages. Any other messages should be considered an error and if a message is unclear it is often caused by a duplicate entry that is already in the cache.

To add an entry to the tunnel cache:

1. With an editor create a file mptnetcsecurityfwmctx.man with the necessary fields described above.
2. On the command line enter: `hand_k mptnetcsecurityfwmctx.man`.

The tunnel context cache entry for the host that is the other end of the tunnel will have the same content with the following lines exchanging values:

```
line 1 and line 2
line 4 and line 6
line 5 and line 7
line 8 and line 11
line 9 and line 12  (Should be the same...no need to swap)
line 10 and line 13
line 14 and line 17 (Should be the same...no need to swap)
line 15 and line 18 (Should be the same...no need to swap)
line 16 and line 19
```

Note: When the other end of a tunnel is either an eNetwork Firewall for AIX, an OS/390 Server, or an AIX V4.3, define a manual tunnel on the partner system and use the export file for the OS/2 configuration. Therefore you don′t have to worry about the other end of the tunnel because that has already been taken care of.

The FWD_K.EXE Command:

FWD_K.EXE is used to delete all the tunnel context entries which match the given source and destination IP addresses from the cache.

To delete all entries in the tunnel cache associated with a set of addresses:

- On the command line enter: `fwd_k <Source IP Address> <Destination IP Address>`. Both the IP addresses MUST be in dotted format, for example, `fwd_k 1.2.3.4 4.3.2.1`.

Tip

We suggest that you use FWD-K.EXE to delete a tunnel context entry before you are updating that entry to avoid problems because all entries are kept in the cache until the system is rebooted. This is just a precaution; we did not discover duplicate tunnel context entries.

The ADMIN.EXE Command:

ADMIN.EXE is used to delete or read one tunnel context entry from the cache. ADMIN takes a file as input with the following format to read a tunnel context entry. (The line has been indented to fit the page.)

```
#  comments
<Source IP address> <Destination IP address> <Context ID> <SAID> <USE_MY_SAID>
   <USE_ESP_SAID>
```

Source IP address IP address of this host. (String: Quad format or alias name.) This is a mandatory field.

Destination IP address IP address of the host at the other end of the tunnel. (String: Quad format or alias name.) This is a mandatory field.

Context ID Tunnel ID. This is an optional field.

SAID Security association ID/security parameter index. This is an optional field.

USE_MY_SAID The character y or Y in this field tells the program to use this host′s SAID instead of remote host′s SAID. This is an optional field.

USE_ESP_SAID The character y or Y in this field tells the program to use security parameter index for encryption rather than security parameter index for authentication. This is an optional field.

For example:

```
   1.2.3.4    4.3.2.1  10  400  y  y
```

If the file contains a line of the above format, the default action of reading the tunnel context is done. If a particular tunnel context is to be deleted, the format of the line is the following. (The line has been indented to fit the page.)

```
@DEL <Source IP Address> <Destination IP address> <Context ID> <SAID>
   <USE_MY_SAID> <USE_ESP_SAID>
```

The DEL command should be preceded by an @ sign and can take any of the three forms DEL, del, or Del.

You can also call ADMIN.EXE without any parameters and then enter the source and destination address of a tunnel to view the parameters from the tunnel context cache. This is essentially the same process as the admin_test utility on the eNetwork Firewall for AIX (see 8.2.2.1, “The admin_test Command” on page 153).

To activate IPSec filter logging, please see 8.5, “OS/2 TCP/IP V4.1 IPSec Client” on page 164.

9.4.2 Interoperability between OS/2 TCP/IP V4.1 IPSec Client and eNetwork Firewall for AIX

Try the following values:

- Tunnel type: Manual
- Tunnel mode: Tunnel
- Policy: Encr/auth (every policy should work)
- AH transform: Keyed MD5
- ESP transform: CDMF

This combination has been tested successfully by us.

Note: We suggest that you export a tunnel definition from the firewall and use those files to determine the corresponding parameters for the OS/2 tunnel context and tunnel policy files. The fwexpmctx.manual file from the firewall would become the fwmctx.man file on OS/2, and the fwexppolicy file from the firewall would become the policy file on OS/2. See 9.4.1.2, “Setting Up IPSec Tunnel Policies” on page 182 and 9.4.1.3, “Setting Up the IPSec Tunnel Context Cache” on page 183 for details on how to modify those files to be used with OS/2.

9.4.3 Interoperability between OS/2 TCP/IP V4.1 IPSec Client and OS/390 Server

Try the following values:

- Tunnel type: Manual
- Tunnel mode: Tunnel
- Policy: Encr/auth (every policy should work)
- AH transform: Keyed MD5
- ESP transform: CDMF

This combination has been tested successfully by us.

Note: We suggest that you export a tunnel definition from the OS/390 Server and use those files to determine the corresponding parameters for the OS/2 tunnel context and tunnel policy files. The fwexpmctx.manual file from the OS/390 Server would become the fwmctx.man file on OS/2, and the fwexppolicy file from the OS/390 Server would become the policy file on OS/2. See 9.4.1.2, "Setting Up IPSec Tunnel Policies" on page 182 and 9.4.1.3, "Setting Up the IPSec Tunnel Context Cache" on page 183 for details on how to modify those files to be used with OS/2.

9.4.4 Interoperability between OS/2 TCP/IP V4.1 IPSec Client and AIX V4.3

Try the following values:

- Tunnel type: Manual
- Tunnel mode: Transport or tunnel
- Policy: Encr/auth (every policy should work)
- AH transform: Keyed MD5
- ESP transform: CDMF
- New header format: No (will be assigned automatically because of the AH transform)
- Replay prevention: No (will be also assigned automatically)

This combination has been tested successfully by us.

Notes:

1. We suggest that you export a tunnel definition from AIX V4.3 and use that file to determine the corresponding parameters for the OS/2 tunnel context and tunnel policy files. The ipsec_tun_manu.exp file from AIX V4.3 would essentially become the fwmctx.man file on OS/2, and the policy file on OS/2 would have to be created by you. See 9.4.1.2, "Setting Up IPSec Tunnel Policies" on page 182 and 9.4.1.3, "Setting Up the IPSec Tunnel Context Cache" on page 183 for details on how to modify that file to be used with OS/2, and see also the following item.
2. It is important that you swap the SPI values for source and destination in a tunnel context file that you derive from an AIX tunnel export file before you commit that tunnel definition on OS/2.
3. If you use policy encr only, please see 6.4.1, "Configuration of the AIX V4.3 Server" on page 120 for details on the required AIX configuration steps.

9.4.5 Interoperability between Two OS/2 TCP/IP V4.1 IPSec Clients

Try the following values:

- Tunnel type: Manual
- Tunnel mode: Transport or tunnel
- Policy: Encr/auth (every policy should work)
- AH transform: Keyed MD5
- ESP transform: CDMF

This combination has been tested successfully by us.

Note: Please see 9.4.1.3, "Setting Up the IPSec Tunnel Context Cache" on page 183 for information on how to modify the fwmctx.man file on the other tunnel endpoint.

9.4.6 Interoperability between OS/2 TCP/IP V4.1 IPSec Client and eNetwork Communications Suite

Try the following values:

- Tunnel type: Manual
- Tunnel mode: Transport
- Policy: auth only (eNetwork Communications Suite does not support CDMF)
- AH transform: Keyed MD5

This combination has been tested successfully by us.

Notes:

1. Do not forget that eNetwork Communications Suite expects SPI values in hexadecimal format.
2. Remember to switch SPI and key values in the fwmctx.man file to make OS/2 a tunnel partner to an eNetwork Communications Suite, as shown in Table 7.

Table 7. Matching Configuration Items between OS/2 TCP/IP V4.1 IPSec Client and eNetwork Communications Suite

OS/2 TCP/IP V4.1 IPSec Client	**eNetwork Communications Suite**
other host IP	local IP
this host IP	remote IP
this host SPI	local SPI
other host SPI	remote SPI
this host key	local key
other host key	remote key
this host transform	local transform
other host transform	remote transform

9.4.7 Turning OS/2 Into a Mini-Firewall

The full IP filtering capabilities of OS/2 TCP/IP V4.1 allow you to set up an OS/2 system as a basic firewall, determining which IP packets to allow to pass between a secure and non-secure network. This capability can be used in conjunction with IPSec or without IPSec. Following are simple instructions on how to configure an OS/2 mini-firewall. We have not tested extensively with that environment, but it has been verified by IBM development.

All interfaces (IP addresses) on the tunnel gateway are assumed to be non-secure. To define an interface as secure the IP address must be added to the file mptnetcfwsecad.cnf on a line by itself. Edit the file mptnetcfwsecad.cnf and add the IP address of your secure interface.

To configure and enable IP filtering, refer to 9.4.1, "Configuring IPSec Filters and Tunnels" on page 180.

In order to allow the OS/2 IPSec gateway to forward IP packets, ipgate must be turned on. To do this enter the following on a command line: `ipgate on`.

Once these three components are correctly configured, filters are active, ipgate is on and routing tables are correctly set up, IP packets should be filtered through a tunnel. IPTRACE and IPFORMAT can be used to see processed packets.

9.5 Additional eNetwork Communications Suite Combinations

This section describes possible VPN scenarios involving the eNetwork Communications Suite that have not been previously discussed.

9.5.1 Interoperability with the eNetwork Firewall for AIX

Configure the firewall like FW1 in the example of Chapter 5, “Branch Office Connection Scenario” on page 81, but choose the following values when defining the tunnel:

1. Tunnel type: Manual
2. Policy: Your choice (but not auth/encr)
3. AH Transform: Keyed MD5
4. ESP Transform: DES CBC 4 or DES CBC 8

Change the filter rules on the firewall as described in 5.7.1, “Allow Only Firewall-to-Firewall Traffic” on page 108. This is necessary, because the filter rules need to allow tunnel traffic for the firewall itself. Export the tunnel definition and print the file fwexpmctx.manual.

On the Windows 95 eNetwork Communications Suite system follow the steps described in 6.4.2, “Configuration of the eNetwork Communications Suite Client” on page 127. Match the values in the export file. (See 9.3.5, “FWTUNNL Export File Formats” on page 177 for an explanation of the contents of the export file.) Remember that the eNetwork Communications Suite expects hexadecimal values for the SPI fields; therefore, you have to convert the decimal firewall values to the corresponding hexadecimal value before entering it on the definition panel.

To make it easier to identify matching configuration values between the two systems, take a look at Table 8.

Table 8. Matching Configuration Items between eNetwork Firewall for AIX and eNetwork Communications Suite

eNetwork Firewall for AIX	eNetwork Communications Suite
owner IP	remote IP
partner IP	local IP
target (GUI) = owner (export file) SPI	remote SPI
firewall (GUI) = partner (export file) SPI	local SPI
owner key	remote key
partner key	local key
owner transform	remote transform
partner transform	local transform

Note: When you use an eNetwork Firewall for AIX V3.2.1 for this combination, pay attention to the fact that the values for target SPI and firewall SPI are reversed by the configuration GUI, as shown in Table 8. To make sure you get the proper SPIs and keys to enter in the eNetwork Communications Suite Security Association panel, look at the fwexpmctx.manual file on the firewall which will have the proper format as described in 9.3.5, “FWTUNNL Export File Formats” on page 177.

See Chapter 6, “Business Partner/Supplier Network Scenario” on page 111 for configuration details if your partner is an AIX V4.3 system. More details on the Windows 95 eNetwork Communications Suite can be found in the redbook *Exploring the IBM eNetwork Communications Suite*, SG24-2111.

Part 4. VPN Future

Chapter 10. The Internet Key Exchange (IKE) Protocol

We have mentioned throughout previous chapters that the crucial elements of IPSec are Security Associations (SA) and the information that they provide in regard of identifying the partners of a secure communications channel, the cryptographic algorithms and keys to be used. You have also learned in 1.3.1.4, "ISAKMP/Oakley" on page 11 that the Internet Key Exchange (IKE) framework, also referred to as ISAKMP/Oakley, supports automated negotiation of Security Associations, and automated generation and refresh of cryptographic keys. The ability to perform these functions with little or no manual configuration of machines will be a critical element as a VPN grows in size.

As a reminder, we would like to repeat that secure exchange of keys is the most critical factor in establishing a secure communications environment. No matter how strong your authentication and encryption are, they are worthless if your key is compromised.

In this chapter, we introduce the concept and basic operation of ISAKMP/Oakley as defined in the current Internet Drafts. A detailed discussion of ISAKMP/Oakley and IBM implementations thereof will be provided in a separate redbook to be published at a later time.

10.1 ISAKMP/Oakley Overview

ISAKMP requires that all information exchanges must be both encrypted and authenticated so that no one can eavesdrop on the keying material, and the keying material will be exchanged only among authenticated parties. This is required because the ISAKMP procedures deal with initializing the keys, so they must be capable of running over links where no security can be assumed to exist. Hence, the ISAKMP protocols use the most complex and processor-intensive operations in the IPSec protocol suite.

In addition, the ISAKMP methods have been designed with the explicit goals of providing protection against several well-known exposures:

- Denial-of-Service: The messages are constructed with unique *cookies* that can be used to quickly identify and reject invalid messages without the need to execute processor-intensive cryptographic operations.
- Man-in-the-Middle: Protection is provided against the common attacks such as deletion of messages, modification of messages, reflecting messages back to the sender, replaying of old messages, and redirection of messages to unintended recipients.
- Perfect Forward Secrecy (PFS): Compromise of past keys provides no useful clues for breaking any other key, whether it occurred before or after the compromised key. That is, each refreshed key will be derived without any dependence on predecessor keys.

10.1.1 The Two Phases of ISAKMP/Oakley

The robustness of any cryptography-based solution depends much more strongly on keeping the keys secret than it does on the actual details of the chosen cryptographic algorithms. Hence, the IETF IPSec Working Group has prescribed a set of extremely robust ISAKMP/Oakley exchange protocols. It uses a 2-phase approach:

1. **Phase 1:** This set of negotiations establishes a master secret from which all cryptographic keys will subsequently be derived for protecting the users′ data traffic. In the most general case, public key cryptography is used to establish an ISAKMP security association between systems, and to establish the keys that will be used to protect the ISAKMP messages that will flow in the subsequent Phase 2 negotiations. Phase 1 is concerned only with establishing the protection suite for the ISAKMP messages themselves, but it does not establish any security associations or keys for protecting user data.

   In Phase 1, the cryptographic operations are the most processor-intensive but need only be done infrequently, and a single Phase 1 exchange can be used to support multiple subsequent Phase 2 exchanges. As a rule of thumb, Phase 1 negotiations are executed once a day or maybe once a week, while Phase 2 negotiations are executed once every few minutes.

2. **Phase 2:** Phase 2 exchanges are less complex, since they are used only after the security protection suite negotiated in Phase 1 has been activated. A set of communicating systems negotiate the security associations and keys that will protect user data exchanges. Phase 2 ISAKMP messages are protected by the ISAKMP security association generated in Phase 1. Phase 2 negotiations generally occur more frequently than Phase 1. For example, a typical application of a Phase 2 negotiation is to refresh the cryptographic keys once every two to three minutes.

An illustration of the use of ISAKMP/Oakley to initially establish security associations and exchange keys between two systems is given in 10.2, “Initializing Security Associations with ISAKMP/Oakley.”

But the ISAKMP protocol offers a solution even when the remote host′s IP address is not known in advance. ISAKMP allows a remote host to identify itself by a *permanent* identifier, such as a name or an e-mail address. The ISAKMP Phase 1 exchanges will then authenticate the remote host′s permanent identity using public key cryptography:

- Certificates create a binding between the permanent identifier and a public key. Therefore, ISAKMP′s certificate-based Phase 1 message exchanges can authenticate the remote host′s permanent identify.
- Since the ISAKMP messages themselves are carried within IP datagrams, the ISAKMP partner (for example, a firewall or destination host) can associate the remote host′s dynamic IP address with its authenticated permanent identity.

See 10.2, “Initializing Security Associations with ISAKMP/Oakley” for a detailed discussion of how ISAKMP/Oakley exchanges authenticate the remote host to its peer and set up the security associations dictated by its corporate VPN policy.

10.2 Initializing Security Associations with ISAKMP/Oakley

This section outlines how ISAKMP/Oakley protocols initially establish security associations and exchange keys between two systems that wish to communicate securely. To provide a concrete example, we describe a message sequence with the following characteristics:

- ISAKMP messages themselves will be carried as a UDP payload.
- ISAKMP Phase 1 exchanges will be authenticated with digital signatures based on certificates obtained from a valid certificate authority.

- Parties participating in the ISAKMP Phase 1 exchanges will be identified by user-based certificates (that is, by name) rather than by host-based certificates (that is, by IP addresses). This is a more general solution, since it can be used even when a host receives a dynamically assigned IP address.
- ISAKMP Phase 2 exchanges will be used to negotiate the protection of user traffic with the ESP protocol, making use of its optional authentication function.

There are other message sequences possible within the ISAKMP framework, but they are not described in this chapter. The interested reader is referred to Appendix B, “Related Publications” on page 209 for technical details.

In the remainder of this section, we assume that the parties involved are named Host-A and Host-B. Host-A will be the initiator of the ISAKMP Phase 1 exchanges, and Host-B will be the responder. If needed for clarity, subscripts A or B will be used to identify the source of various fields in the message exchanges.

10.2.1 Phase 1 - Setting Up ISAKMP/Oakley Security Associations

The security associations that protect the ISAKMP messages themselves are set up during the Phase 1 exchanges. Since we are starting ″cold″ (no previous keys or SAs have been negotiated between Host-A and Host-B), the Phase 1 exchanges will use the ISAKMP Identity Protect exchange (also known as Oakley Main Mode). Six messages are needed to complete the exchange:

- Messages 1 and 2 negotiate the characteristics of the security associations. Messages 1 and 2 flow in the clear for the initial Phase 1 exchange, and they are unauthenticated.
- Messages 3 and 4 exchange nonces and also execute a Diffie-Hellman exchange to establish a master key (SKEYID). Messages 3 and 4 flow in the clear for the initial Phase 1 exchange, and they are unauthenticated.
- Messages 5 and 6 exchange digital signatures, and optionally the pertinent user-based certificates for the purpose of mutually authenticating the parties′ identities. The payloads of Messages 5 and 6 are protected by the encryption algorithm and keying material established with messages 1 through 4.

10.2.1.1 ISAKMP Phase 1, Message 1

Since Host-A is the initiating party, it will construct a cleartext ISAKMP message (Message 1) and send it to Host-B. The ISAKMP message itself is carried as the payload of a UDP packet, which in turn is carried as the payload of a normal IP datagram (see Figure 103 on page 196).

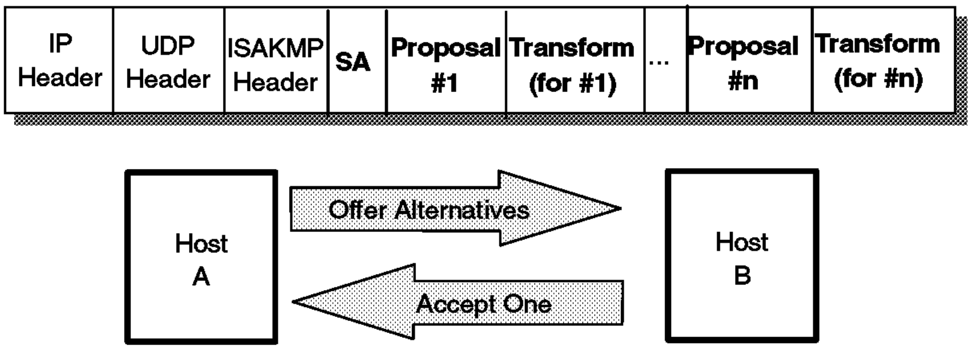


Figure 103. Message 1 of an ISAKMP Phase 1 Exchange

The source and destination addresses to be placed in the IP header are those of Host-A (initiator) and Host-B (responder), respectively. The UDP header will identify that the destination port is 500, which has been assigned for use by the ISAKMP protocol. The payload of the UDP packet carries the ISAKMP message itself.

In Message 1, Host-A, the initiator, proposes a set of one or more protection suites for consideration by Host-B, the responder. Hence, the ISAKMP Message contains at least the following fields in its payload:

- The ISAKMP Header in Message 1 will indicate an exchange type of Main Mode, and will contain a Message ID of 0. Host-A will set the Responder Cookie field to 0, and will fill in a random value of its choice for the Initiator Cookie, which we will denote as Cookie-A.
- The Security Association field identifies the Domain of Interpretation (DOI). Since the hosts plan to run IPSec protocols between themselves, the DOI is simply IP.
- Host-A′s Proposal Payload will specify the protocol PROTO_ISAKMP and will set the SPI value to 0.

  Note: For ISAKMP Phase 1 messages, the actual SPI field within the Proposal Payload is not used to identify the ISAKMP Security Association. During Phase 1, the ISAKMP SA is identified instead by the pair of values <Initiator Cookie, Responder Cookie>, both of which must be non-zero values. Since the Responder Cookie has not yet been generated by Host-B, the ISAKMP SA is not yet unambiguously identified.

- The Transform Payload will specify KEY_OAKLEY. For the KEY_OAKLEY transform, Host-A must also specify the relevant attributes: namely, the authentication method to be used, the pseudo-random function to be used, and the encryption algorithm to be used. Host-A will specify: authentication using digital signatures, a pseudo-random function of HMAC-MD5, and an encryption algorithm of DES-CBC.

10.2.1.2 ISAKMP Phase 1, Message 2

In Message 1, Host-A proposed one or more candidate protection suites to be used to protect the ISAKMP exchanges. Host-B uses Message 2 to indicate which one, if any, it will support. Note that in our example, Host-A proposed just a single option, so Host-B merely needs to acknowledge that the proposal is acceptable.

The message contents will be as follows:

- The source and destination addresses to be placed in the IP header are those of Host-B (responder) and Host-A (initiator), respectively. The UDP header will identify that the destination port is 500, which has been assigned for use by the ISAKMP protocol. The payload of the UDP packet carries the ISAKMP message itself.
- The ISAKMP Header in Message 2 will indicate an exchange type of Main Mode, and will contain a Message ID of 0. Host-B will set the Responder Cookie field to a random value, which we will call Cookie-B, and will copy into the Initiator Cookie field the value that was received in the Cookie-A field of Message 1. The value pair <Cookie-A, Cookie-B> will serve as the SPI for the ISAKMP Security Association.
- The Security Association field identifies the Domain of Interpretation (DOI). Since the hosts plan to run IPSec protocols between themselves, the DOI is simply IP.
- Host-B′s Proposal Payload will specify the protocol PROTO_ISAKMP and will set the SPI value to 0.

  Note: For ISAKMP Phase 1 messages, the actual SPI field within the Proposal Payload is not used to identify the ISAKMP Security Association. During Phase 1, the ISAKMP SA is identified instead by the pair of values <Initiator Cookie, Responder Cookie>, both of which must be non-zero values.

- The Transform Payload will specify KEY_OAKLEY. For the KEY_OAKLEY transform, the attributes that were accepted from the proposal offered by Host-A are copied into the appropriate fields. That is, Host-B will confirm that it will use: authentication using digital signatures, a pseudo-random function of HMAC-MD5, and an encryption algorithm of DES-CBC.

Properties of ISAKMP SA Established

At this point, the properties of the ISAKMP Security Association have been agreed to by Host-A and Host-B. The identity of the ISAKMP SA has been set equal to the pair <Cookie-A, Cookie-B>. However, the identities of the parties claiming to be Host-A and Host-B have not yet been authoritatively verified.

10.2.1.3 ISAKMP Phase 1, Message 3

The third message of the Phase 1 ISAKMP exchange begins the exchange of the information from which the cryptographic keys will eventually be derived (see Figure 104 on page 198). All information is exchanged in the clear. None of the messages themselves carry the actual cryptographic keys. Instead, they carry inputs that will be used by Host-A and Host-B to derive the keys locally. The ISAKMP payload will be used to exchange two types of information:

- Diffie-Hellman public value: g^x from the initiator. The exponent x in the public value is the private value that must be kept secret.
- Nonce: N_i from the initiator. (*Nonce* is a fancy name for a value that is considered to be random according to some very strict mathematical guidelines.)

These values are carried in the Key Exchange field and the Nonce field.

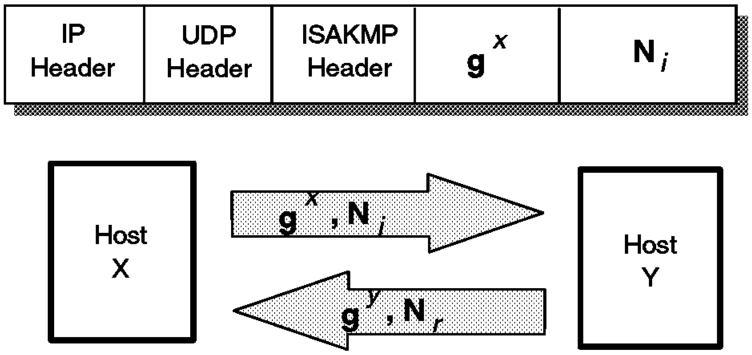


Figure 104. Message 3 of an ISAKMP Phase 1 Exchange

10.2.1.4 ISAKMP Phase 1, Message 4

After receiving a Diffie-Hellman public value and a nonce from Host-A, Host-B will respond by sending to Host-A its own Diffie-Hellman public value (g^y from the responder) and its nonce (N_r from the responder).

10.2.1.5 Generating the Keys (Phase 1)

At this point, each host knows the values of the two nonces (N_i and N_r). Each host also knows its own private Diffie-Hellman value (x and y) and also knows its partner′s public value (g^x or g^y). Hence each side can construct the composite value g^{xy}. And finally, each side knows the values of the initiator cookie and the responder cookie.

Given all these bits of information, each side can then independently compute identical values for the following quantities:

- SKEYID: This collection of bits is sometimes referred to as keying material, since it provides the raw input from which actual cryptographic keys will be derived later in the process. It is obtained by applying the agreed-to pseudorandom function (in our example, HMAC-MD5) to the known inputs:

  SKEYID = HMAC-MD5(N_i, N_r, g^{xy})

- Having computed the value SKEYID, each side then proceeds to generate two cryptographic keys and some additional keying material:
  - SKEYID_d is keying material that will be subsequently used in Phase 2 to derive the keys that will be used in non-ISAKMP SAs for protecting user traffic:

    SKEYID_d = HMAC-MD5(SKEYID, g^{xy}, CookieA, CookieB, 0)

  - SKEYID_a is the key used for authenticating ISAKMP messages:

    SKEYID_a = HMAC-MD5(SKEYID, SKEYID_d, g^{xy}, CookieA, CookieB, 1)

  - SKEYID_e is the key used for encrypting ISAKMP exchanges:

    SKEYID_e = HMAC-MD5(SKEYID, SKEYID_a, g^{xy}, CookieA, CookieB, 2)

Keys are Available

At this point in the protocol, both Host-A and Host-B have derived identical authentication and encryption keys that they will use to protect the ISAKMP exchanges. And they have also derived identical keying material from which they will derive keys to protect user data during Phase 2 of the ISAKMP negotiations. However, at this point, the two parties′ identities still have not been authenticated to one another.

10.2.1.6 ISAKMP Phase 1, Message 5

At this point in the Phase 1 flows, the two hosts will exchange identity information with each other, using the Digital Signature Algorithm to authenticate themselves. As shown in Figure 105, the ISAKMP message will carry an identity payload, a signature payload, and an optional certificate payload. Host-A uses Message 5 to send information to Host-B that will allow Host-B to authenticate Host-A.

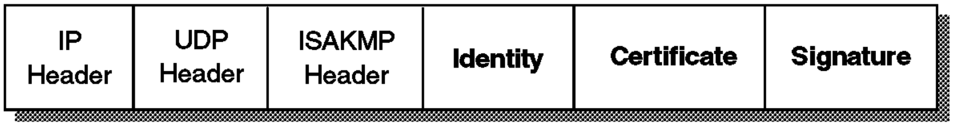


Figure 105. Message 5 of an ISAKMP Phase 1 Exchange

When an actual certificate is present in the Certificate Payload field, the receiver can use the information directly, after verifying that it has been signed with a valid signature of a trusted certificate authority. If there is no certificate in the message, then it is the responsibility of the receiver to obtain a certificate using some implementation method. For example, it may send a query to a trusted certificate authority using a protocol such as LDAP, or it may query a secure DNS server, or it may maintain a secure local cache that maps previously used certificates to their respective ID values, or it may send an ISAKMP Certificate Request message to its peer, who must then immediately send its certificate to the requester.

Note: The method for obtaining a certificate is a local option, and is not defined as part of ISAKMP/Oakley. In particular, it is a local responsibility of the receiver to check that the certificate in question is still valid and has not been revoked.

There are several points to bear in mind:

- At this stage of the process, all ISAKMP payloads, whether in Phase 1 or Phase 2, are now encrypted, using the encryption algorithm (negotiated in Messages 1 and 2) and the keys (derived from the information in Messages 3 and 4). The ISAKMP header itself, however, is still transmitted in the clear.
- In Phase 1, IPSec′s ESP protocol is not used: that is, there is no ESP header. The recipient uses the Encryption Bit in the Flags field of the ISAKMP header to determine if encryption has been applied to the message. The pair of values <CookieA, CookieB>, which serve as an SPI for Phase 1 exchanges, provide a pointer to the correct algorithm and key to be used to decrypt the message.
- The Digital Signature is not applied to the ISAKMP message itself. Instead, it is applied to a hash of information that is available to both Host-A and Host-B.

- The identity carried in the identity payload does not necessarily bear any relationship to the source IP address; however, the identity carried in the identity payload must be the identity to which the certificate applies.

Since the pseudo-random function for the ISAKMP SA is HMAC-MD5 in our example, Host-A (the initiator) will generate the following hash function, sign it using the Digital Signature Algorithm, and then place the result in the Signature Payload field:

$$\text{HASH\_I} = \text{HMAC-MD5}(\text{SKEYID}, g^x, g^y, \text{CookieA}, \text{CookieB}, SA_p, ID_A)$$

ID_A is Host-A′s identity information that was transmitted in the identity payload of this message, and SA_p is the entire body of the SA payload that was sent by Host-A in Message 1, including all proposals and all transforms proposed by Host-A. The cookies, public Diffie-Hellman values, and SKEYID were explicitly carried in Messages 1 through 4, or were derived from their contents.

10.2.1.7 ISAKMP Phase 1, Message 6

After receiving Message 5 from Host-A, Host-B will verify the identity of Host-A by validating the digital signature. If the signature is valid, then Host-B will send Message 6 to Host-A to allow Host-A to verify the identity of Host-B.

The structure of Message 6 is the same as that of Message 5, with the obvious changes that the identity payload and the certificate payload now pertain to Host-B. A less obvious difference is that the hash that is signed by Host-B is different from the one previously signed by Host-A:

$$\text{HASH\_R} = \text{HMAC-MD5}(\text{SKEYID}, g^y, g^x, \text{CookieB}, \text{CookieA}, SA_p, ID_B)$$

Notice that the order in which Diffie-Hellman public values and the cookies appear has been changed, and the final term now is the Identity Payload that Host-B has included in Message 6.

Phase-1 is Complete

When Host-A receives Message 6 and verifies the digital signature, the Phase 1 exchanges are then complete. At this point, each participant has authenticated itself to its peer, both have agreed on the characteristics of the ISAKMP Security Associations, and both have derived the same set of keys (or keying material).

10.2.1.8 Miscellaneous Phase 1 Facts

There are several miscellaneous facts worth noting:

1. The Phase 1 message flows above pertain to the case where the ISAKMP/Oakley messages will be authenticated by the Digital Signature Standard. There are other permissible ways to authenticate ISAKMP messages. For example, pre-shared keys or public key encryption could also be used. Regardless of the specific authentication mechanism that is used, there will be six messages exchanged for Oakley Main Mode. However, the content of the individual messages will differ, depending on the authentication method. The key calculation formula for SKEYID will also differ depending on the authentication method.

2. Although ISAKMP/Oakley exchanges make use of both encryption and authentication, they do not use either IPSec′s ESP or AH protocol. ISAKMP exchanges are protected with application-layer security mechanisms, not with network layer security mechanisms.
3. ISAKMP messages are sent using UDP. There is no guaranteed delivery for them.
4. The only way to identify that an ISAKMP message is part of a Phase 1 flow rather than a Phase 2 flow is to check the Message ID field in the ISAKMP Header. For Phase 1 flows, it must be 0, and (although not explicitly stated in the ISAKMP documents) for Phase 2 flows it must be non-zero.

10.2.2 Phase 2 - Setting Up Non-ISAKMP/Oakley Security Associations

After having completed the Phase 1 negotiation process to set up the ISAKMP Security Associations, Host-A′s next step is to initiate the ISAKMP/Oakley Phase 2 message exchanges (also known as Oakley Quick Mode) to define the security associations and keys that will be used to protect IP datagrams exchanged between the pair of users. (In the Internet Drafts, these are referred to somewhat obtusely as ″non-ISAKMP SAs″).

Because the purpose of the Phase 1 negotiations was to agree on how to protect ISAKMP messages, all ISAKMP Phase 2 payloads, but not the ISAKMP header itself, that must be encrypted using the algorithm agreed to by the Phase 1 negotiations.

When Oakley Quick Mode is used in Phase 2, authentication is achieved via the use of several cryptographically-based hash functions. The input to the hash functions comes partly from Phase 1 information (SKEYID) and partly from information exchanged in Phase 2. Phase 2 authentication is based on certificates, but the Phase 2 process itself does not use certificates directly. Instead, it uses the SKEYID_a material from Phase 1, which itself was authenticated via certificates.

Oakley Quick Mode comes in two forms:

- Without a Key Exchange attribute, Quick Mode can be used to refresh the cryptographic keys, but does not provide the property of Perfect Forward Secrecy (PFS).
- With a Key Exchange attribute, Quick Mode can be used to refresh the cryptographic keys in a way that provides PFS. This is accomplished by including an exchange of public Diffie-Hellman values within messages 1 and 2.

Note: PFS apparently is a property that is very much desired by cryptography experts, but strangely enough, the specs treat PFS as ″optional″. They mandate that a system must be capable of handling the Key Exchange field when it is present in a Quick Mode message, but do not require a system to include the field within the message.

An overview of the three messages within Quick Mode is given below. In our example, we have Host-A propose two protection suites, and then illustrate how Host-B chooses just one of them:

1. Proposal 1 will offer protection via ESP with DES-CBC for encryption and HMAC-MD5 for authentication.
2. Proposal 2 will offer protection via AH with HMAC-MD5 for authentication.

10.2.2.1 ISAKMP Phase 2, Message 1

Message 1 of a Quick Mode Exchange allows Host-A to authenticate itself, to select a nonce, to propose security association(s) to Host-B, to execute an exchange of public Diffie-Hellman values, and to indicate if it is acting on its own behalf or as a proxy negotiator for another entity. An overview of the format of Message 1 is shown in Figure 106.

Note: Inclusion of a key exchange field is optional. However, when Perfect Forward Secrecy is desired, it must be present.

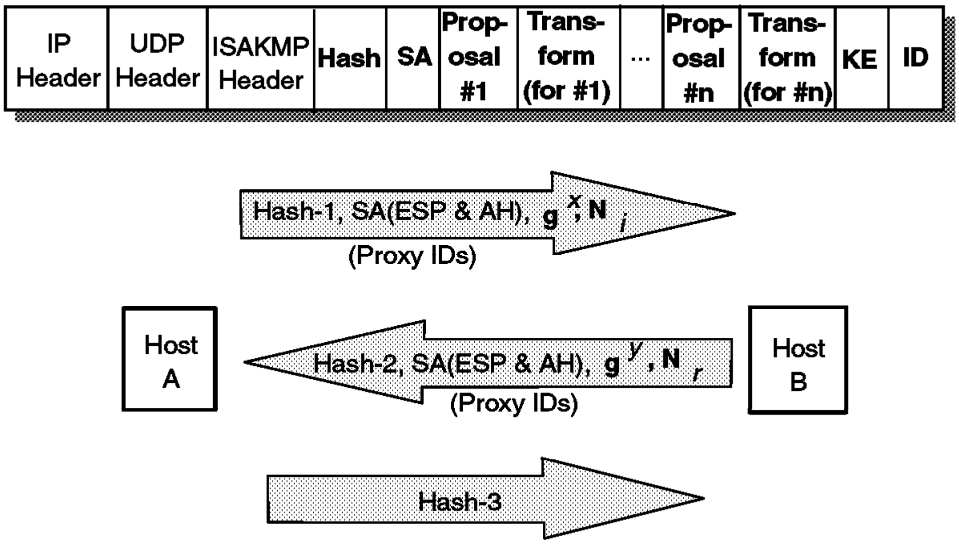


Figure 106. Message 1 of an ISAKMP Phase 2 Quick Mode Exchange

Since we have assumed that Host-A and Host-B are each acting on their own behalf, the user identity fields illustrated in Figure 106 will not be present. The message will consist of

- ISAKMP Header: Will indicate an exchange type of Quick Mode, will include a non-zero Message-ID chosen by Host-A, will include the initiator and responder cookie values chosen in Phase 1 (that is, Cookie-A and Cookie-B), and will turn on the encryption flag to indicate that the payloads of the ISAKMP message are encrypted according to the algorithm and key negotiated during Phase 1.
- HASH_1: A Hash Payload must immediately follow the ISAKMP header. HASH_1 uses the pseudo-random function that was negotiated during the Phase 1 exchanges; in our example, this is HMAC-MD5, thus:

  HASH_1 = HMAC-MD5(SKEYID_a, M-ID, SA, N_i, KE)

  HASH_1 is derived from the following information:

  - SKEYID_a was derived from the Phase 1 exchanges.
  - M-ID is the Message-ID of this message.
  - SA is the Security Association payload carried in this message, including all proposals that were offered.
  - KE is the public Diffie-Hellman value carried in this message. This quantity is chosen by Host-A, and is denoted as g_{qm}^x. Note that this is not the same quantity as g^x that was used in the Phase 1 exchanges.
- Security Association Payload: Indicate IP as the Domain of Interpretation.

- Proposal, Transform Pairs: There will be two of these pairs in this example:
  1. First Protection Suite: The first proposal payload will be numbered 1, will identify ESP as the protocol to be used, and will include an SPI value that is randomly chosen by Host-A for use with the ESP protocol. The proposal payload will be followed by a single transform payload that indicates ESP_DES as the algorithm.

     Note: In the current Domain of Interpretation document, the code point ESP_DES calls for both encryption with DES-CBC and authentication with HMAC-MD5.
  2. Second Protection Suite: The second proposal payload will be numbered 2, will identify AH as the protocol to be used, and will include an SPI value that is randomly chosen by Host-A for use with the AH protocol. The proposal payload will be followed by a single transform payload that names HMAC-MD5 as the algorithm.
- Nonce: This contains the nonce N^i that was chosen randomly by Host-A.
- KE: This is the key exchange payload that will carry the public Diffie-Hellman value chosen by Host-A, g_{qm}^x. There is also a field called Group, that indicates the prime number and generator used in the Diffie-Hellman exchange.

10.2.2.2 ISAKMP Phase 2, Message 2

After Host-B receives Message 1 from Host-A and successfully authenticates it using HASH_1, it constructs a reply, Message 2, to be sent back to Host-A. The Message ID of the reply will be the same one that Host-A used in Message 1. Host-B will choose new values for the following:

- Nonce payload now carries N_r, a random value chosen by Host-B.
- Key exchange payload now carries Host-B′s public Diffie-Hellman value, g_{qm}^y.
- Hash payload now carries the value HASH_2, which is defined as:

$$\text{HASH\_2} = \text{HMAC-MD5}(\text{SKEYID\_a}, N_i, \text{M-ID}, \text{SA}, N_r, \text{KE})$$

- Security payload only describes the single chosen proposal and its associated transforms, not all of the protection suites offered by Host-A. In this example, Host-B will select Proposal #1, which will use ESP_DES (encryption plus authentication) as the protocol. Host-B also chooses an SPI value for the ESP_DES, the selected protocol; Host-B′s SPI does not depend in any way on the SPI that Host-A assigned to that protocol when it offered the proposal. That is, it is not necessary that SPI_A be the same as SPI_B; it is only necessary that they each be non-zero and that they each be randomly chosen.

Keys for Non-ISAKMP Can Be Derived

At this point, Host-A and Host-B have exchanged nonces and public Diffie-Hellman values. Each one can use this in conjunction with other information to derive a pair of keys, one for each direction of transmission.

10.2.2.3 Generating the Keys (Phase 2)

Using the nonces, public Diffie-Hellman values, SPIs, protocol code points exchanged in Messages 1 and 2 of Phase 2, and the SKEYID value from Phase 1, each host now has enough information to derive two sets of keying material.

- For data generated by Host-A and received by Host-B, the keying material is:

$$KEYMAT_{AB} = HMAC\text{-}MD5(SKEYID, {g_{qm}}^{xy}, protocol, SPI_B, N_i, N_r)$$

- For data generated by Host-B and received by Host-A, the keying material is:

$$KEYMAT_{BA} = HMAC\text{-}MD5(SKEYID, {g_{qm}}^{xy}, protocol, SPI_A, N_i, N_r)$$

Note: In this example, Host-A needs to derive four keys:

1. Key for generating the integrity check value for transmitted datagrams
2. Key for validating the integrity check value of received datagrams
3. Key for encrypting transmitted datagrams
4. Key for decrypting received datagrams

Likewise, Host-B needs to derive the mirror image of the same four keys. For example, the key that Host-B uses to encrypt its outbound messages is the same key that Host-A uses to decrypt its inbound messages, etc.

10.2.2.4 ISAKMP Phase 2, Message 3

At this point, Host-A and Host-B have exchanged all the information necessary for them to derive the necessary keying material. The third message in the Quick Mode exchange is used by Host-A to prove its liveness, which it does by producing a hash function that covers the Message ID and both nonces that were exchanged in Messages 1 and 2. Message 3 consists only of the ISAKMP header and a hash payload that carries:

$$HASH\_3 = HMAC\text{-}MD5(SKEYID\_a, 0, M\text{-}ID, N_i, N_r)$$

When Host-B receives this message and verifies the hash, then both systems can begin to use the negotiated security protocols to protect their user data streams.

10.3 Negotiating Multiple Security Associations

The example covered in 10.2.2.1, “ISAKMP Phase 2, Message 1” on page 202 through 10.2.2.4, “ISAKMP Phase 2, Message 3” illustrated a case where a single non-ISAKMP security association was negotiated by means of a Quick Mode message exchange. However, it is also possible to negotiate multiple security associations, each with its own set of keying material, within a single 3-message Quick Mode exchange.

The message formats are very similar to the previously illustrated ones, so only the differences will be highlighted below:

- Message 1 will carry multiple security association payloads, each offering a range of protection suites.

- HASH_1 will cover the entire set of all offered Security Associations carried in Message 1. That is, each Security Association and all of its offered proposals are included.
- In Message 2, for each offered SA, Host-B will select a single protection suite. That is, if n SAs are open for negotiation, then Host-B will choose n protection suites, one from each proposal.
- As was the case for HASH_1, HASH_2 will now cover the entire set of all offered security associations carried in Message 1. That is, each security association and all of its offered proposals are included.
- After Messages 1 and 2 have been exchanged, then Host-A and Host-B will be able to generate the keying material for each of the accepted protection suites, using the same formulas as in 10.2.2.3, "Generating the Keys (Phase 2)" on page 204, applied individually for each accepted SA. Even though the nonces and the public Diffie-Hellman values are the same for all selected suites, the keying material derived for each selected protection suite will be different because each proposal will have a different SPI.
- Because multiple security associations have been negotiated, it is a matter of local choice as to which one is used to protect a given datagram. A receiving system must be capable of processing a datagram that is protected by any SA that has been negotiated. That is, it would be legal for a given source host to send two consecutive datagrams to a destination system, where each datagram was protected by a different SA.

10.4 Using ISAKMP/Oakley with Remote Access

The critical element in the remote access scenario is the use of ISAKMP/Oakley to identify the remote host by name, rather than by its dynamically assigned IP address. Once the remote host′s identity has been authenticated and the mapping to its dynamically assigned IP address has been ascertained, the remainder of the processes are the same as we have described for the other scenarios. For example, if the corporate intranet is considered to be trusted, then the remote host needs to establish a single SA between itself and the firewall. But if the corporate intranet is considered to be untrusted, then it may be necessary for the remote host to set up two SAs: one between itself and the firewall, and a second between itself and the destination host.

Recall that a single ISAKMP Phase 1 negotiation can protect several subsequent Phase 2 negotiations. Phase 1 ISAKMP negotiations use computationally intensive public key cryptographic operations, while Phase 2 negotiations use the less computationally intensive symmetric key cryptographic operations. Hence, the heavy computational load only occurs in Phase I, which will only be executed once when the dial-up connection is first initiated.

The principal points that pertain to the remote access case are:

- The remote host′s dynamically assigned address is the one that is placed in the IP header of all ISAKMP messages.
- The remote host′s permanent identifier (such as an e-mail address, for example) is the quantity that is placed in the ID field of the ISAKMP Phase 1 messages.
- The remote host′s certificate used in the ISAKMP exchange must be associated with the remote host′s permanent identifier.

- In traffic-bearing datagrams, the remote host′s dynamically assigned IP address will be used. This is necessary since the destination IP address that appears in the datagram′s IP header is used in conjunction with the SPI and protocol type to identify the relevant IPSec security association for processing the inbound datagram.

Appendix A. Special Notices

This redbook is targeted at networking engineers and administrators to provide them with the necessary background information as well as hands-on product examples to successfully deploy VPNs. It is also intended for I/T consultants to identify the benefits of VPNs in general and the IBM eNetwork VPN solutions in particular. The information in this publication is not intended as the specification of any programming interfaces that are provided by any of the products discussed herein. See the PUBLICATIONS section of the IBM Programming Announcement for the discussed products for more information about what publications are considered to be product documentation.

References in this publication to IBM products, programs or services do not imply that IBM intends to make these available in all countries in which IBM operates. Any reference to an IBM product, program, or service is not intended to state or imply that only IBM′s product, program, or service may be used. Any functionally equivalent program that does not infringe any of IBM′s intellectual property rights may be used instead of the IBM product, program or service.

Information in this book was developed in conjunction with use of the equipment specified, and is limited in application to those specific hardware and software products and levels.

IBM may have patents or pending patent applications covering subject matter in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to the IBM Director of Licensing, IBM Corporation, 500 Columbus Avenue, Thornwood, NY 10594 USA.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact IBM Corporation, Dept. 600A, Mail Drop 1329, Somers, NY 10589 USA.

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The information contained in this document has not been submitted to any formal IBM test and is distributed AS IS. The information about non-IBM (″vendor″) products in this manual has been supplied by the vendor and IBM assumes no responsibility for its accuracy or completeness. The use of this information or the implementation of any of these techniques is a customer responsibility and depends on the customer′s ability to evaluate and integrate them into the customer′s operational environment. While each item may have been reviewed by IBM for accuracy in a specific situation, there is no guarantee that the same or similar results will be obtained elsewhere. Customers attempting to adapt these techniques to their own environments do so at their own risk.

Any pointers in this publication to external Web sites are provided for convenience only and do not in any manner serve as an endorsement of these Web sites.

Any performance data contained in this document was determined in a controlled environment, and therefore, the results that may be obtained in other operating environments may vary significantly. Users of this document should verify the applicable data for their specific environment.

Reference to PTF numbers that have not been released through the normal distribution process does not imply general availability. The purpose of including these reference numbers is to alert IBM customers to specific information relative to the implementation of the PTF when it becomes available to each customer according to the normal IBM PTF distribution process.

The following terms are trademarks of the International Business Machines Corporation in the United States and/or other countries:

AIX®
eNetwork
IBM®
LANStreamer®
OS/2®
OS/400®
SecureWay

AS/400®
IBM Global Network
LAN Distance®
Nways
OS/390
PowerPC®

The following terms are trademarks of other companies:

C-bus is a trademark of Corollary, Inc.

Java and HotJava are trademarks of Sun Microsystems, Incorporated.

Microsoft, Windows, Windows NT, and the Windows 95 logo are trademarks or registered trademarks of Microsoft Corporation.

PC Direct is a trademark of Ziff Communications Company and is used by IBM Corporation under license.

Pentium, MMX, ProShare, LANDesk, and ActionMedia are trademarks or registered trademarks of Intel Corporation in the U.S. and other countries.

UNIX is a registered trademark in the United States and other countries licensed exclusively through X/Open Company Limited.

Other company, product, and service names may be trademarks or service marks of others.

Appendix B. Related Publications

The publications listed in this section are considered particularly suitable for a more detailed discussion of the topics covered in this redbook.

B.1 International Technical Support Organization Publications

For information on ordering these ITSO publications see “How to Get ITSO Redbooks” on page 211.

- *Protect and Survive Using IBM Firewall 3.1 for AIX*, SG24-2577
- *Exploring the IBM eNetwork Communications Suite*, SG24-2111
- *AIX Version 4.3 Differences Guide*, SG24-2014
- *Secure Electronic Transactions: Credit Card Payment on the Web in Theory and Practice*, SG24-4978

B.2 Redbooks on CD-ROMs

Redbooks are also available on CD-ROMs. **Order a subscription** and receive updates 2-4 times a year at significant savings.

CD-ROM Title	**Subscription Number**	**Collection Kit Number**
System/390 Redbooks Collection	SBOF-7201	SK2T-2177
Networking and Systems Management Redbooks Collection	SBOF-7370	SK2T-6022
Transaction Processing and Data Management Redbook	SBOF-7240	SK2T-8038
Lotus Redbooks Collection	SBOF-6899	SK2T-8039
Tivoli Redbooks Collection	SBOF-6898	SK2T-8044
AS/400 Redbooks Collection	SBOF-7270	SK2T-2849
RS/6000 Redbooks Collection (HTML, BkMgr)	SBOF-7230	SK2T-8040
RS/6000 Redbooks Collection (PostScript)	SBOF-7205	SK2T-8041
RS/6000 Redbooks Collection (PDF Format)	SBOF-8700	SK2T-8043
Application Development Redbooks Collection	SBOF-7290	SK2T-8037

B.3 Other Publications

These publications are also relevant as further information sources:

- *eNetwork Firewall for AIX, User*, GC31-8419
- *eNetwork Firewall for AIX, Reference Guide*, SC31-8418
- *Applied Cryptography*, second edition, John Wiley & Sons, Inc., 1996, by Bruce Schneier; ISBN 0-471-11709-9.
- *Network Security: Private Communication in a Public World*, PTR Prentice Hall, 1995, by Charlie Kaufman, Radia Perlman, and Mike Speciner; ISBN 0-13-061466-1.
- *Request For Comments (RFC)* and *Internet Drafts (ID)*

  There are more than 2300 RFCs today. For those readers who want to keep up-to-date with the latest advances and research activities in TCP/IP, the ever-increasing number of RFCs and Internet Drafts (ID) is the best source of this information. RFCs can be viewed or obtained online from the Internet Engineering Taskforce (IETF) Web page using the following URL: `http://www.ietf.org`.

B.3.1 Web Site Reference

The following is a list of World Wide Web sites that we also consider as relevant sources of further information:

Current IPSec Internet Drafts (see3.5, "Current IPSec Internet Drafts" on page 55 for a detailed list of the core IPSec specifications)
  http://www.ietf.org/ids.by.wg/ipsec.html

IBM eNetwork home page
  http://www.software.ibm.com/enetwork

IBM eNetwork Firewall home page
  http://www.software.ibm.com/enetwork/firewall

IBM eNetwork Virtual Private Networks home page
  http://www.software.ibm.com/enetwork/technology/vpn/

IBM AIX home page
  http://www.rs6000.ibm.com/software/aix_os.html

IBM OS/390 IPSec information
  http://www.s390.ibm.com/products/mvs/firewall/ipsec.htm

IBM Software Choice home page (TCP/IP V4.1 for OS/2 and prerequisites)
  http://service.software.ibm.com/asd-bin/doc/index.htm

IBM Corp. home page
  http://www.ibm.com

FTP Software, Inc., home page
  http://www.ftp.com

SSL Information
  http://home.netscape.com/assist/security/ssl/index.html

U.S. version of Pretty Good Privacy (PGP)
  http://www.pgp.com.

International version of Pretty Good Privacy (PGP)
  http://www.pgpi.com.

How to Get ITSO Redbooks

This section explains how both customers and IBM employees can find out about ITSO redbooks, CD-ROMs, workshops, and residencies. A form for ordering books and CD-ROMs is also provided.

This information was current at the time of publication, but is continually subject to change. The latest information may be found at `http://www.redbooks.ibm.com/`.

How IBM Employees Can Get ITSO Redbooks

Employees may request ITSO deliverables (redbooks, BookManager BOOKs, and CD-ROMs) and information about redbooks, workshops, and residencies in the following ways:

- **Redbooks Web Site on the World Wide Web**

  `http://w3.itso.ibm.com/`

- **PUBORDER** — to order hardcopies in the United States
- **Tools Disks**

  To get LIST3820s of redbooks, type one of the following commands:

  ```
  TOOLCAT REDPRINT
  TOOLS SENDTO EHONE4 TOOLS2 REDPRINT GET SG24xxxx PACKAGE
  TOOLS SENDTO CANVM2 TOOLS REDPRINT GET SG24xxxx PACKAGE (Canadian users only)
  ```

  To get BookManager BOOKs of redbooks, type the following command:

  ```
  TOOLCAT REDBOOKS
  ```

  To get lists of redbooks, type the following command:

  ```
  TOOLS SENDTO USDIST MKTTOOLS MKTTOOLS GET ITSOCAT TXT
  ```

  To register for information on workshops, residencies, and redbooks, type the following command:

  ```
  TOOLS SENDTO WTSCPOK TOOLS ZDISK GET ITSOREGI 1998
  ```

- **REDBOOKS Category on INEWS**
- **Online** — send orders to: USIB6FPL at IBMMAIL or DKIBMBSH at IBMMAIL

Redpieces

For information so current it is still in the process of being written, look at ″Redpieces″ on the Redbooks Web Site (`http://www.redbooks.ibm.com/redpieces.html`). Redpieces are redbooks in progress; not all redbooks become redpieces, and sometimes just a few chapters will be published this way. The intent is to get the information out much quicker than the formal publishing process allows.

How Customers Can Get ITSO Redbooks

Customers may request ITSO deliverables (redbooks, BookManager BOOKs, and CD-ROMs) and information about redbooks, workshops, and residencies in the following ways:

- **Online Orders** — send orders to:

	IBMMAIL	**Internet**
In United States:	usib6fpl at ibmmail	usib6fpl@ibmmail.com
In Canada:	caibmbkz at ibmmail	lmannix@vnet.ibm.com
Outside North America:	dkibmbsh at ibmmail	bookshop@dk.ibm.com

- **Telephone Orders**

United States (toll free)	1-800-879-2755
Canada (toll free)	1-800-IBM-4YOU
Outside North America	(long distance charges apply)
(+45) 4810-1320 - Danish	(+45) 4810-1020 - German
(+45) 4810-1420 - Dutch	(+45) 4810-1620 - Italian
(+45) 4810-1540 - English	(+45) 4810-1270 - Norwegian
(+45) 4810-1670 - Finnish	(+45) 4810-1120 - Spanish
(+45) 4810-1220 - French	(+45) 4810-1170 - Swedish

- **Mail Orders** — send orders to:

IBM Publications
Publications Customer Support
P.O. Box 29570
Raleigh, NC 27626-0570
USA

IBM Publications
144-4th Avenue, S.W.
Calgary, Alberta T2P 3N5
Canada

IBM Direct Services
Sortemosevej 21
DK-3450 Allerød
Denmark

- **Fax** — send orders to:

United States (toll free)	1-800-445-9269
Canada	1-403-267-4455
Outside North America	(+45) 48 14 2207 (long distance charge)

- **1-800-IBM-4FAX (United States)** or **(+1)001-408-256-5422 (Outside USA)** — ask for:

Index # 4421 Abstracts of new redbooks
Index # 4422 IBM redbooks
Index # 4420 Redbooks for last six months

- **On the World Wide Web**

Redbooks Web Site	http://www.redbooks.ibm.com/
IBM Direct Publications Catalog	http://www.elink.ibmlink.ibm.com/pbl/pbl

Redpieces

For information so current it is still in the process of being written, look at ″Redpieces″ on the Redbooks Web Site (`http://www.redbooks.ibm.com/redpieces.html`). Redpieces are redbooks in progress; not all redbooks become redpieces, and sometimes just a few chapters will be published this way. The intent is to get the information out much quicker than the formal publishing process allows.

IBM Redbook Order Form

Please send me the following:

Title	**Order Number**	**Quantity**

First name Last name

Company

Address

City Postal code Country

Telephone number Telefax number VAT number

- Invoice to customer number ____________________

- Credit card number ____________________

Credit card expiration date Card issued to Signature

We accept American Express, Diners, Eurocard, Master Card, and Visa. Payment by credit card not available in all countries. Signature mandatory for credit card payment.

List of Abbreviations

AH	Authentication Header
AIX	Advanced Interactive Executive
API	Application Programming Interface
ARP	Address Resolution Protocol
ASCII	American Standard Code for Information Interchange
AS/400	Application System/400
ATM	Asynchronous Transfer Mode
CA	Certification Authority
CBC	Cipher Block Chaining
CHAP	Challenge Handshake Authentication Protocol
CPU	Central Processing Unit
CDMF	Commercial Data Masking Facility
DDNS	Dynamic Domain Name System
DES	Digital Encryption Standard
DHCP	Dynamic Host Configuration Protocol
DLL	Dynamic Link Library
DMZ	De-Militarized Zone
DNS	Domain Name Server
DOI	Domain of Interpretation
DOS	Disk Operating System
DSA	Digital Signature Algorithm
DSS	Digital Signature Standard
EBCDIC	Extended Binary Communication Data Interchange Code
ESP	Encapsulating Security Payload
FTP	File Transfer Protocol
GUI	Graphical User Interface
HMAC	Hashed Message Authentication Code
HTML	Hypertext Markup Language
HTTP	Hypertext Transfer Protocol
IAB	Internet Activities Board
IANA	Internet Assigned Numbers Authority
IBM	International Business Machines Corporation
ICMP	Internet Control Message Protocol
ICSS	Internet Connection Secure Server
ICV	Integrity Check Value
IDEA	International Data Encryption Algorithm
IEEE	Institute of Electrical and Electronics Engineers
IESG	Internet Engineering Steering Group
IETF	Internet Engineering Task Force
IGMP	Internet Group Management Protocol
IGN	IBM Global Network
IKE	Internet Key Exchange
IP	Internet Protocol
IPC	Interprocess Communication
IPSec	IP Security Architecture
ISAKMP	Internet Security Association and Key Management Protocol
ISDN	Integrated Services Digital Network
ISO	International Standards Organization
ISP	Internet Service Provider
ITSO	International Technical Support Organization
IV	Initialization Vector
LAN	Local Area Network
LDAP	Lightweight Directory Access Protocol
LLC	Logical Link Layer
L2TP	Layer 2 Tunneling Protocol
MAC	Message Authentication Code
MAC	Media Access Control
MD2	RSA Message Digest 2 Algorithm
MD5	RSA Message Digest 5 Algorithm
MIB	Management Information Base

MIME	Multipurpose Internet Mail Extensions
MPTN	Multiprotocol Transport Network
MS-CHAP	Microsoft Challenge Handshake Authentication Protocol
MVS	Multiple Virtual Storage Operating System
NAT	Network Address Translation
NDIS	Network Device Interface Specification
NFS	Network File System
NIC	Network Information Center
NIS	Network Information Systems
NIST	National Institute of Standards and Technology
NNTP	Network News Transfer Protocol
NSA	National Security Agency
NTP	Network Time Protocol
NVT	Network Virtual Terminal
OSPF	Open Shortest Path First
OS/2	Operating System/2
PAP	Password Authentication Protocol
PGP	Pretty Good Privacy
POP	Post Office Protocol
PPP	Point-to-Point Protocol
PPTP	Point to Point Tunneling Protocol
PSTN	Public Switched Telephone Network
QOS	Quality of Service
RAM	Random Access Memory
RARP	Reverse Address Resolution Protocol
RAS	Remote Access Service
RC4	RSA Rivest Cipher 4 Algorithm
RFC	Request for Comments
RISC	Reduced Instruction Set Computer
RIP	Routing Information Protocol
ROM	Read-only Memory
RSH	Remote Shell
RS/6000	IBM RISC System/6000
SA	Security Association
SET	Secure Electronic Transactions
S-HTTP	Secure Hypertext Transfer Protocol
SLIP	Serial Line Internet Protocol
S-MIME	Secure Multipurpose Internet Mail Extension
SMIT	System Management Interface Tool
SMTP	Simple Mail Transfer Protocol
SNG	Secured Network Gateway (former product name of the AIX firewall)
SNMP	Simple Network Management Protocol
Socks	SOCK-et-S: An internal NEC development name that remained after release
SPI	Security Parameter Index
SSL	Secure Sockets Layer
TCP	Transmission Control Protocol
TCP/IP	Transmission Control Protocol / Internet Protocol
TOS	Type of Service
TTL	Time to Live
UDP	User Datagram Protocol
URL	Uniform Resource Locator
VPN	Virtual Private Network
WAN	Wide Area Network
WWW	World Wide Web
3DES	Triple Digital Encryption Standard

Index

ITSO Redbook Evaluation

A Comprehensive Guide to Virtual Private Networks, Volume I:
SG24-5201-00

Your feedback is very important to help us maintain the quality of ITSO redbooks. **Please complete this questionnaire and return it using one of the following methods:**

- Use the online evaluation form found at http://www.redbooks.ibm.com
- Fax this form to: USA International Access Code + 1 914 432 8264
- Send your comments in an Internet note to redbook@us.ibm.com

Which of the following best describes you?
__Customer   __Business Partner   __Independent Software Vendor   __IBM employee
__None of the above

Please rate your overall satisfaction with this book using the scale:
(1 = very good, 2 = good, 3 = average, 4 = poor, 5 = very poor)

Overall Satisfaction ____________

Please answer the following questions:

Was this redbook published in time for your needs? Yes____ No____

If no, please explain:
____

____

____

____

What other redbooks would you like to see published?
____

____

____

Comments/Suggestions: (THANK YOU FOR YOUR FEEDBACK!)
____

____

____

____

____

A Comprehensive Guide to Virtual Private Networks, Volume I:
IBM Firewall, Server and Client Solutions

SG24-5201-00

SG24-5201-00
Printed in the U.S.A.