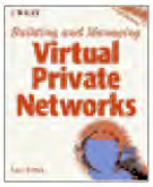
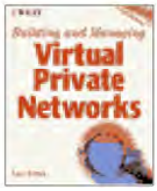


Building and Managing Virtual Private Networks
by Dave Kosiur
Wiley Computer Publishing, John Wiley & Sons, Inc.
ISBN: 0471295264 **Pub Date:** 09/01/98



Building and Managing Virtual Private Networks
by Dave Kosiur
Wiley Computer Publishing, John Wiley & Sons, Inc.
ISBN: 0471295264 **Pub Date:** 09/01/98

Preface

The world of *virtual private networks* (VPNs) has exploded in the last year, with more and more vendors offering what they call VPN solutions for business customers. Unfortunately, each vendor has his own definition of what a VPN is; to add to the confusion, each potential customer has his own idea of what comprises a VPN as well. Mix in the usual portion of marketing hype, and you've got quite a confusing situation indeed.

One of the purposes of this book is to dispell as much of the confusion surrounding VPNs as possible. Our approach has been based on three main ideas: relate the current usage of the term VPN to past private networks so that both experienced and new network managers can see how they're related; carefully describe and compare the various protocols so that you, the reader, will see the advantages and disadvantages of each; and always keep in mind that more than one kind of VPN fits into the business environment. With the wide variety of technologies available for VPNs, it should be the customer who decides what kind of VPN—and, therefore, what protocols and products—meets his business needs best.

To that end, this book aims to provide you with the background on VPN technologies and products that you need to make appropriate business decisions about the design of a VPN and expectations for its use.

Who Should Read This Book

This book is aimed at business and IS managers, system administrators, and network managers who are looking to understand what Internet-based VPNs are and how they can be set up for business use. Our goal is to provide the reader with enough background to understand the concepts, protocols, and systems associated with VPNs so that his company can decide whether it wants to deploy a VPN and what might be the best way to do so, in terms of cost, performance, and technology.

How This Book Is Organized

This book has been organized into five parts:

1. The Internet and Business
2. Securing an Internet VPN
3. Building Blocks of a VPN
4. Managing a VPN
5. Looking Ahead

Part I, *The Internet and Business,* covers the relationship between business and Internet, including how VPNs can provide competitive advantages to businesses. The first three chapters of the book make up Part I.

Chapter 1, "Business on the Internet," discusses today's current dynamic business environment, the basics of the Internet, and how Internet technology meshes with business needs using intranets, extranets, and VPNs.

Chapter 2, "Virtual Private Networks," covers the different types of private networks and *virtual private networks* (VPNs) that have been deployed by businesses over the past 30 years and introduces the focus of this book, virtual private networks created using the Internet. Here, you'll find details on cost justifications for Internet-based VPNs, along with other reasons for using VPNs.

Chapter 3, "A Closer Look at Internet VPNs," delves into the nature of Internet-based VPNs, introducing their architecture as well as the components and protocols that can be used to create a VPN over the Internet.

Part II, *Securing an Internet VPN,* focuses on the security threats facing Internet users and how the three main VPN protocols—IPSec, PPTP, and L2TP—deal with these security issues so that you can properly design a VPN to meet your needs. Chapters 4 through 8 are included in Part II.

Chapter 4, "Security: Threats and Solutions," describes the major threats to network security and then moves on to detail the principles of different systems for authenticating users and how cryptography is used to protect your data.

Chapter 5, "Using IPSec to Build a VPN," is the first of three chapters presenting the details of the main protocols used to create VPNs over the Internet. The first of the trio covers the *IP Security Protocol* (IPSec) and the network components you can use with IPSec for a VPN.

Chapter 6, "Using PPTP to Build a VPN," discusses the details of PPTP, the Point-to-Point Tunneling Protocol. Like Chapter 5, it includes a discussion of protocol details and the devices that can be deployed to create a VPN.

Chapter 7, "Using L2TP to Build a VPN," is the last chapter dealing with VPN protocols; it covers L2TP, the Layer2 Tunneling Protocol. It shows how L2TP incorporates some of the features of PPTP and IPSec and how its VPN devices differ from those of the other two protocols.

Chapter 8, "Designing Your VPN," focuses on the issues you should deal with in planning your VPN. The major considerations you'll most likely face in VPN design are classified into three main groups—network issues, security issues, and ISP issues. This chapter aims to serve as a transition from many of the theoretical and protocol-related issues discussed in the first seven chapters of the book to the more pragmatic issues of selecting products and deploying and managing the VPN, which is the focus of the remainder of the book.

Part III, *Building Blocks of a VPN,* moves into the realm of the products that are available for creating VPNs, as well as the role the ISP can play in your VPN.

Chapter 9, "The ISP Connection," focuses on Internet Service Providers, showing how they relate to the Internet's infrastructure and the service you can expect from them. Because your VPN is likely to become mission-critical, the role of the ISP is crucial to the VPN's success. We, therefore, cover how

service level agreements are used to state expected ISP performance and how they can be monitored. The last part of this chapter summarizes some of the current ISPs that offer special VPN services, including outsourced VPNs.

Chapter 10, "Firewalls and Routers," is the first of three chapters that deal with VPN products. This chapter discusses how firewalls and routers can be used to create VPNs. For each type of network device, we cover the principal VPN-related requirements and summarize many of the products that are currently available in the VPN market.

Chapter 11, "VPN Hardware," continues the product coverage, focusing on VPN hardware. One main issue covered in the chapter is the network services that should be integrated in the hardware and the resulting effects on network performance and management.

Chapter 12, "VPN Software," deals with VPN software, mainly the products that can be used with existing servers or as adjuncts to Network Operating Systems. As in the previous two chapters, this chapter includes a list of requirements and a summary of the available products.

Part IV, *Managing a VPN,* includes three chapters that cover the three main issues of management—security, IP addresses, and performance.

Chapter 13, "Security Management," describes how VPNs have to mesh with corporate security policies and the new policies that may have to be formulated, particularly for managing cryptographic keys and digital certificates. The chapter includes suggestions on selecting encryption key lengths, deploying authentication services, and how to manage a certificate server for digital certificates.

Chapter 14, "IP Address Management," covers some of the problems network managers face in allocating IP addresses and naming services. It describes the solutions using *Dynamic Host Configuration Protocol* (DHCP) and *Dynamic Domain Name System* (DDNS) and points out some of the problems VPNs can cause with private addressing, *Network Address Translation* (NAT), and DNS.

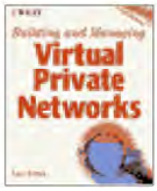
Chapter 15, "Performance Management," is concerned with the basics of network performance and how the demands of new network applications like interactive multimedia can be met both on networks and VPNs. The chapter describes the five major approaches to providing differentiated services and how network management can be tied to VPN devices, especially through policy-based network management.

Part V, the last part of the book, is called *Looking Ahead* and covers likely ways to expand your VPN and what the future may hold.

Chapter 16, "Extending VPNs to Extranets," deals specifically with the issues of extending your VPN to become an extranet to link business partners together for electronic commerce. It covers some of the main reasons for creating an extranet and points out some of the issues you'll have to deal with while getting all the parts of an extranet to work together.

Chapter 17, "Future Directions," is our attempt to project where the VPN market is going and what's likely to happen in the next few years, in the development of VPN protocols, the products that support them, and the uses businesses will create for VPNs.

Previous | Table of Contents | Next



Building and Managing Virtual Private Networks
by Dave Kosiur
Wiley Computer Publishing, John Wiley & Sons, Inc.
ISBN: 0471295264 **Pub Date:** 09/01/98

PART I
The Internet and Business

Virtual Private Networks (VPNs) now can provide cost savings of 50 to 75 percent by replacing more costly leased lines and remote access servers and reducing equipment and training costs; but they also help keep your business network flexible, enabling it to respond faster to changes in business partnerships and the marketplace.

As you evaluate your corporate structure for designing a VPN, keep in mind which sites require full-time connections and what type of data will cross the VPN, as well as how many telecommuters and mobile workers you'll need to support.

CHAPTER 1
Business on the Internet

Communication is the heart of business. Not only do companies depend on communication to run their internal affairs, but they also have to communicate with their suppliers, customers, and markets if they expect to stay in business.

In the 90s, the Internet has become the star of communication. It has captured the imaginations of individuals and business owners alike as a new medium for communicating with customers as well as business partners. But, the Internet is a great melting pot of many different technologies. Many of the technologies necessary for reliable, secure business quality communications are still in the process of being rolled out for routine use. The everyday use of the Internet for business communication holds great promise, but we've yet to achieve the plug-and-play stage for many business applications of the Internet.

Today's advances in technology at every level of networking can make it difficult, if not impossible, to find a single integrated solution for your business needs. Thus, we find ourselves in the midst of a time in which not only are new higher-speed media being introduced for residential and business communication, but in which new application environments, such as the Web, not only unify diverse services but offer added opportunities such as the new marketing and sales channels found in electronic commerce.

The terminology surrounding the Internet seems to change every day as vendors seek to define new market niches and offer their versions of "marketectures." One aim of this book is to address the

confusion surrounding the technologies that fall under the umbrella term *Virtual Private Networks* (VPNs), providing you with a framework for distinguishing between the different types of VPNs and selecting the ones that will meet your business needs.

This book focuses on running VPNs over the Internet. Using the Internet for a Virtual Private Network enables you to communicate securely among your offices—wherever they may be located—with greater flexibility and at a lower cost than using private networks set up with pre-Internet technologies, such as leased lines and modem banks.

This chapter serves as a brief introduction to the structure and capabilities of today's Internet and how the Internet can be used by businesses to improve their operations. Later chapters will cover the details of many of the concepts we introduce here.

The Changing Business Environment

Business today isn't like it was in the good old days, even if old is only 3–5 years ago. Amidst all the downsizing, automation, and increasing numbers of small businesses as well as mega-mergers, one trend seems self-evident: Flexibility is the order of the day.

A cornerstone of business flexibility is an adaptable communications network. Well-designed networking can help your business deal with many of the changes in current-day business environments—for example, improved customer and partner relations, an increasingly mobile workforce, flattened organizational structures, virtual teams, etc. (see Figure 1.1).

Businesses are faced not only with quickly changing projects and markets but also with short-term associations with suppliers and other business partners as they attempt to compete. Customers demand more—not just more quality and variety in products but also more information about, and support for, the products. As customers demand more, they also can offer more to sellers; smart marketers look to increased interactivity with customers to learn more of their needs, leaning towards more individuality and treating each customer as a market of one rather than a large number of individuals lumped into a single group with average tastes and needs.

FIGURE 1.1 Changes in today's business environments.

Even as businesses struggle with these sources and sinks of information, they find their own employees dispersed across the planet, trying to get their jobs done in markets that have become increasingly global. Businesspersons may well hope that phone calls and videoconferences can make the deal or solve a problem, but we're still stuck in a physical world in which face-to-face contacts are valued, useful, and often a necessity. Thus, we're faced with an increasingly mobile workforce, and I'm not referring to job-switching (although that happens often enough), just to the number of miles the modern-day worker travels to meet business obligations. Yet, amidst all this travel across the planet, each employee needs to stay in touch with the home office, wherever it is.

One of the common business trends in the past decade has been a flattening of the business organization,

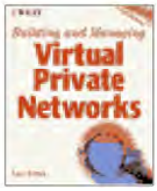
a move from a hierarchical management structure to one including fewer managers and more interacting teams. Flatter organizations, however, require more coordination and communication in order to function properly, providing yet another reason for the growth of networks.

In these flatter organizations, it's not uncommon to see an increasing number of teams formed. These teams, which are formed quickly to attack a particular problem and then disbanded, consist of members scattered throughout the company, often in more than one country. Much of their work and coordination is conducted electronically, transmitted across networks at any and all times of the day. In a global business, the sun never sets.

As businesses change, so too must the Information Technology (IT) departments helping to maintain the communication infrastructure that's so important to the company's success. Three major shifts in information technology have occurred during the past few years—from personal computing to workgroup computing, from islands of isolated systems to integrated systems, and from intra-enterprise computing to inter-enterprise computing. To deal with all these changes and help synchronize the organization with business, the IT staff have to maintain flexibility so they can respond to the regular order of the day—change.

A primary aim of this book is to illustrate how the Internet and *Internet Protocol* (IP)-based technologies can provide your business with new methods for creating a more flexible and less costly private network that better meets today's business needs. Let's investigate the Internet a bit before we move on to the details of these Internet-based Virtual Private Networks.

Previous | Table of Contents | Next



Building and Managing Virtual Private Networks
by Dave Kosiur
Wiley Computer Publishing, John Wiley & Sons, Inc.
ISBN: 0471295264 **Pub Date:** 09/01/98

The Internet

In spite of all the hype and heightened expectations surrounding it, the Internet has truly become one of the major technological achievements of this century. Starting out as a simple network connecting four computers scattered around the United States, the Internet has become the largest public data network, crisscrossing the globe and connecting peoples of all ages, nationalities, and ways of life. Even as it's become a common mode of communication among individuals using computers at home and at the workplace, the Internet has become more of a commercial network, offering businesses new forms of connectivity, both with other business partners and with their customers.

For all its success, the Internet can be difficult for some to fathom. For instance, the Internet has no central governing body that can compel its users to follow a particular procedure. A number of organizations deal with different aspects of the Internet's governance. For instance, the *Internet Society* (ISOC) helps promote policies and the global connectivity of the Internet, while the *Internet Engineering Task Force* (IETF) is a standards setting body for many of the technical aspects. The *World Wide Web Consortium* (W3C) focuses on standards for the Web and interacts with the IETF in setting standards. Addressing and naming of entities on the Internet is important to the functioning of the Internet, and that task currently is shared by Network Solutions Inc. and the *Internet Assigned Numbers Authority* (IANA), although the parties involved in this procedure may change before long.

The Internet is a somewhat loose aggregation of networks that work together by virtue of running according to a common set of rules, or protocols, the *Transfer Control Protocol/Internet Protocol* (TCP/IP) protocols. These protocols have proven to be an important cornerstone of the Internet, which has evolved in a very open environment guided by a group of selfless, dedicated engineers under the guidance of the *Internet Architecture Board* (IAB), the overseer of the IETF, and a related task force, the *Internet Research Task Force* (IRTF). Despite the proliferation of numerous other networking protocols, the TCP/IP protocols have become the preferred means for creating open, extensible networks, both within and among businesses as well as for public networking. The seemingly never-ending exponential growth of the Internet that started roughly three decades ago is but one proof of the Internet's popularity and flexibility.

The growth of the Internet has been phenomenal by any measure (see Figure 1.2). The Internet's predecessor, ARPANET, was started in 1969 and connected only four computers at different locations in the United States. During the past few years, the number of computers attached to the Internet has been doubling annually. According to the survey of Internet domains that's been run periodically since 1987 by Network Wizards, more than 30 million computers were connected to the Internet as of February, 1998. Depending on whom you ask, 50 million users of the Internet may live in the United States alone.

With this growth has come a change in the direction of the Internet. Although the Internet may have started out as a network designed primarily for academic research, it's now become a commercialized network frequented largely by individuals outside universities and populated by a large number of business enterprises.

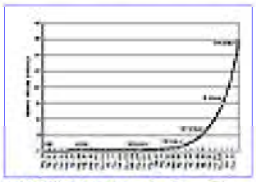


FIGURE 1.2 Growth of the Internet.

Business usage of the Internet has grown as well. It's difficult to measure business-related traffic in any reliable coherent fashion. But, one sample indicator of phenomenal growth of business use is the increase in the number of computers in what are called *.com domain names* (reserved for businesses only)—the number of these business-related computers rose from 774,735 in July, 1994, to 8,201,511 in August, 1997.

The Internet's Infrastructure

The Internet is global in scope and strongly decentralized with no single governing body. The physical networks comprising the Internet form a hierarchy (see Figure 1.3) whose top level is composed of the high-speed backbone network maintained by MCI (now part of Worldcom); the majority of Internet traffic is funnelled onto the backbone through the *Network Access Points* (NAPs), which are maintained by Sprint, Worldcom, and others—these are located in strategic metropolitan areas across the United States (see Figure 1.4).

Independently-created national networks set up by PSInet and UUNET, among others, mostly tie into the NAPs, but some service providers have made their own arrangements for peering points to help relieve some of the load at the NAPs. Lower levels are composed of regional networks, then the individual networks found on university campuses, at research organizations, and in businesses.

For most users, the internal structure of the Internet is transparent. They connect to the Internet via their *Internet Service Provider* (ISP) and send e-mail, browse the Web, share files, and connect to other host computers on the Internet without concern for where those other computers are located or how they're connected to the Internet. We'll cover some of the details of tying your internal networks to the Internet in the following chapters.

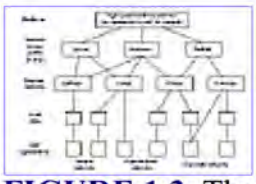


FIGURE 1.3 The Internet hierarchy.

What the Internet Delivers

For a moment, put aside any specific business needs that you may have. Instead, just concentrate on what the Internet can offer its users.

The Internet offers its users a wide range of connectivity options, many at low cost. These options range from a very high-speed (megabits per second) direct link to the Internet backbone to support data exchange or multimedia applications between company sites to the low-end option of using a dial-up connection through regular phone lines at speeds of 9,600 to 28,800 bits per seconds.

The near-ubiquity of the Internet makes setting up connections much easier than with any other data network. These could be either permanent connections for branch offices or on-the-fly links for your mobile workers. While Internet coverage isn't equal throughout the world, the Internet makes it possible to achieve global connectivity at a cost lower than if your business created its own global network.

As mentioned before, the Internet is built on a series of open protocols. This foundation has made it much easier for developers to write networked applications for just about any computing platform, promoting a great deal of interoperability. It's not unusual to find a wide range of Internet applications that run on all major operating systems, making your job of offering common networked services easier. The World Wide Web has gone even farther by offering developers and content designers alike the possibility of working within a single user interface that spans multiple operating systems as well.

FIGURE 1.4 Map of U.S. Internet.

The Internet also offers you the opportunity of having a more manageable network. Because you've outsourced much of the national and global connectivity issues to your Internet Service Provider, you can focus more of your attention on other internal network management issues.

Previous | Table of Contents | Next

The Internet is not without its shortcomings, however. In many ways, it's become a victim of its own success. For example, the bandwidth available on the Internet backbone and offered by many ISPs has barely been able to keep up with the explosive increase in Internet usage that's taken place during the past few years. That, in turn, has raised some concerns about the reliability of Internet traffic. Brownouts and other localized network outages have occurred, but new equipment and policies continue to improve the robustness of Internet links.

A related concern has been the Internet's capability to handle multimedia traffic, especially real-time interactive multimedia. In general, the delays of data transmissions over the Internet make real-time multimedia transmissions difficult, but certain ISP networks have been designed with such applications in mind, and efforts at improving quality-of-service have started to address the problem. Currently, guaranteed performance is restricted by most ISPs to network uptime, but you should expect to see minimum delay guarantees offered in the next year or two.

Lastly, and this is an issue we'll repeatedly address in this book, is the problem of security. Admittedly, the majority of data transmitted on the Internet is transmitted in the clear and can be intercepted by others. But, methods exist for encrypting data against illegal viewing as well as for preventing unauthorized access to private corporate resources, even when they're linked to the Internet. Many of the reported illegal intrusions into networks are due more to poorly-implemented security policies than to any inherent insecurity of the Internet. We'll see later in this book that robust security is available for every aspect of data communications over the Internet.

Using Internet Technology

The Internet offers business opportunities on what we'll call a *private level* as well as a *public level*. The public level is where a great deal of attention has been focused over the past few years, as proponents of electronic commerce have aimed at the buying and selling of goods and services over the public Internet, either to the general public or to other businesses.

But, the private Internet is what this book is all about. Businesses can use the Internet as a means of transmitting corporate information privately among their corporate sites, without fear that either hackers or the general public will see the information. The plumbing and many of the techniques are the same for both the public Internet and private businesses using the Internet, but the goal differs—open data for public access versus protected, private data for businesses. We'll see in this book that the two goals are not contradictory nor are they mutually exclusive.

The fact that these two uses can share many of the same telecommunications resources offers new opportunities for business (see Figure 1.5).

Moving private business data on the Internet can also simplify, or at least ease, the setup of more business-to-business opportunities. The commonality of the Internet—its protocols, plumbing, the popular Web interface, and so on—make it easier to ensure compatibility between two or more business partners (if they've embraced the use of the Internet). If you're already distributing private business data on the Internet to a select group of employees, it's not difficult to expand the membership of that select group to include a new corporate partner. Today's techniques make setting up links between new business partners a matter of days, if not hours—as long as you're on the Internet.

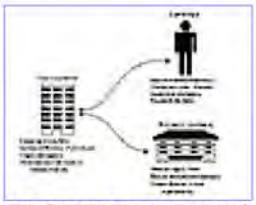


FIGURE 1.5 Using the Internet for business.

The openness of the TCP/IP protocols and the interoperability that the protocols promote hasn't escaped the attention of the business world. Now we're seeing not only increased usage of that grand-daddy of TCP/IP networks, the Internet (with a capital I), but more and more businesses are using TCP/IP to create their own corporate networks or intranets, tying together disparate technologies and different types of computers into intranets. Now the same applications and expertise that have been used on the Internet can be deployed within corporate networks for their own private uses.

It seems only natural that, if your company's using TCP/IP for its internal networks and if you want to communicate with business partners, suppliers, and the like (who are also using TCP/IP), the Internet can become the link between your business and theirs. This underlying concept of extranets means that you control access to your computing resources and your business partner does likewise for his resources, but you use TCP/IP over the Internet to share common data and increase the efficiency of communications between the two of you (see Figure 1.6).

We'll return to extranets later. The majority of this book is going to focus on another aspect of TCP/IP networks for business, using the Internet to link together a company's sites and mobile workers into one private, secure network. VPNs make secure multisite intranets possible. While intranets primarily focus on a set of applications, notably the Web, within a corporate organization, VPNs provide the lower-layer network services (or plumbing). Extranets also have a focus on applications that's similar to that found in intranets, but they're between business partners. VPNs also make extranets easier to implement, because the security services offered by VPNs enable you to control access to your corporate resources, and that access can include business partners and suppliers.

Internet-based VPNs, the subject of this book, enable you to leverage many of the Internet's inherent advantages—global connectivity, distributed resources, and location-independence, for example—to add value to your business's internal operations (see Figure 1.7). Not only can you save money and improve connections to international business partners, but you can support more flexible working arrangements, both for your employees and business partners.

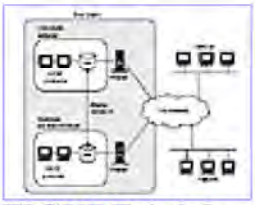


FIGURE 1.6 Intranets, extranets, and VPNs.

FIGURE 1.7 Using the Internet's capabilities to improve business.

Summary

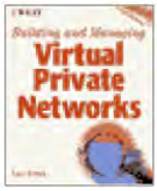
Much of today's business is focused on information—its creation, analysis, or distribution. This preoccupation with information as a source of revenue and competitive advantage not only drives the exchange of information between workers and teams within a company but also drives the exchange of information between business partners as well as between businesses and their customers.

Today's accompanying focus on computers and things digital dovetails nicely with the demand for more and more information. Digital information is so much easier to obtain and distribute via electronic means that networks are becoming both the circulatory and nervous systems of the business world.

While private networks have long proven their usefulness in many corporate environments, the current-day trend to obtain information from a multitude of sources, many of them outside the corporate walls, has business managers and network architects alike looking for ways to tie together their internal private electronic networks with external, more public ones.

The Internet offers businesses the means to improve communications not only with their customers and business partners but also with other parts of the company. Creating secure, private corporate networks using the shared infrastructure of the Internet is what the remainder of this book is about.

Previous | Table of Contents | Next



Building and Managing Virtual Private Networks
by Dave Kosiur
Wiley Computer Publishing, John Wiley & Sons, Inc.
ISBN: 0471295264 **Pub Date:** 09/01/98

CHAPTER 2
Virtual Private Networks

Ever since businesses started to use computers in more than one location, there's been the desire and the need to connect them together in a private, secure fashion to facilitate corporate communications. Setting up a private network on a local campus of office buildings can be relatively simple, because the company usually owns the physical plant. But, installing a corporate network involving other offices or plants located miles away in another county or state makes things more difficult. In many cases, businesses have had no choice but to use special phone lines leased from their local exchange or long-distance carriers in order to link together geographically separated locations.

You'll see as we go through the following section that businesses have long had various ways to interconnect their sites, forming private corporate networks. But, until recently, these networks were essentially *hard-wired*, offering little flexibility. After network services were offered to connect sites over shared public links, the term *Virtual Public Network* or VPN became part of the vernacular. The word "virtual" was tacked on as a modifier to indicate that although you could treat the circuit between two sites as a private one, it was, in fact, not hard-wired and existed only as a link when traffic was passing over the circuit. *It was a virtual circuit.* As we see later in this chapter, a major concern when setting up virtual circuits for transmitting private data on Internet VPNs is protecting that data from illegal interception and unauthorized viewing.

The Evolution of Private Networks

During the past 30 plus years, the nature and architecture of private corporate networks have evolved as new technologies have become available and business environments have changed. What started out as private networks using phone lines leased from AT&ampT have now become virtual private networks using the Internet as the primary communications medium.

If you were to trace corporate networking back to the 1960s, you would see that business managers had little choice but to connect their sites using analog phone lines and 2,400-bps modems leased from AT&ampT. Eventually, as the telephone monopoly and government policies changed, other companies pushed modem technology forward, enabling businesses to link their sites at higher speeds, reaching 9,600 bps in the early '80s.

Although we may be accustomed to the idea of using a laptop and a modem just about anywhere we go these days, many modem-based links 30 years ago were statically-defined links between stationary sites,

not the dynamic mobile ones of today. The best quality analog lines were specially-selected ones, called *conditioned lines*, that were permanently wired to a site; there also weren't that many mobile workers running around with portable computers and modems.

For most, the leased lines used for intersite corporate connectivity were dedicated circuits that connected two endpoints on a network (see Figure 2.1). The dedicated circuits were not switched via the *public switched telephone network* (PSTN) like regular phone calls but were configured for full-time use by a single party—the corporate customer. The bandwidth of that circuit was dedicated to the customer's use and was not shared with other customers. The advantage of this architecture is that the customer is guaranteed both bandwidth and privacy on the line. One disadvantage is that the customer must pay for the full bandwidth on the line at all times, even when the line is not being used.

Although these networks were private, in that they consisted of point-to-point connections over lines devoted just to the client's traffic, these networks couldn't be called virtual private networks, because more than one customer of the network provider (i.e., the phone companies) didn't share the transmission media. VPNs were to come later.

FIGURE 2.1 A private network of leased lines.

The next significant advance for connecting sites came with the introduction of *Digital Data Service* (DDS) in the mid 1970s. DDS was the first digital service for private line applications, offering 56-Kbps connections to corporate customers.

As digital services became more readily available, interest in *Wide Area Networks* (WANs) using these services grew. Connections using T1 services running at 1.544 Mbps were particularly useful. A T1 datastream consists of 24 separate channels, each of which can carry up to 64 Kbps of traffic (called a DSO stream or channel), either voice or data. Because these channels could be assigned to different uses, a company could use a single T1 line to service both its voice and data networking needs, assigning different numbers of channels to each use according to its internal requirements.

Defining the VPN

Many different definitions of Virtual Private Network are floating around the marketplace; many of these definitions have been tweaked to meet the product lines and focus of the vendors. We've settled on one rather simple definition for VPNs that we'll use throughout this book—a *Virtual Private Network is a network of virtual circuits for carrying private traffic.*

A virtual circuit is a connection set up on a network between a sender and a receiver in which both the route for the session and bandwidth is allocated dynamically. VPNs can be established between two or more Local Area Networks (LANs), or between remote users and a LAN.

In the early 1990s, the driving force for private networks was voice communications, not data. Phone companies traditionally sold T1 services to corporate clients as a way to create their own lower cost private telephone systems, pointing out that the cost savings of this approach to voice communications

enabled clients to let data traffic between sites piggy-back on the otherwise unused bandwidth of the T1 links.

But, as markets changed and the cost of voice communications through the telcos dropped, the cost savings of private voice networks disappeared, or at least was greatly reduced. At the same time, data traffic had increased, and interest in using either T1s or 56-Kbps lines for mainly data traffic grew.

During the past few years, other networking technologies like frame relay and *Asynchronous Transfer Mode* (ATM) have become available for forming corporate networks. Frame relay has become particularly popular for connecting different sites together. Less equipment is needed at each endpoint, because a router at each endpoint can take care of directing the traffic to more than one destination (see Figure 2.3 on page 22). That's because the service provider maintains a "cloud" of frame relay connections, and the links are assigned only as needed.

Because the frame-relay links are assigned only when needed, frame relay corporate nets probably are the first modern-day virtual private networks. (It's worth noting that X.25 packet-switched networks also used virtual circuits and used *Closed User Groups* [CUGs] to restrict recipients of data. The X.25 networks probably also should be classified as VPNs, but newer technologies like frame-relay appear to be deployed more frequently these days.)

Although this frame-relay net can simplify connections somewhat when compared to the mesh of leased lines because you need to connect only each site to the provider's frame-relay cloud and although it offers less expensive connectivity than leased lines, the frame-relay net does not address the needs of mobile workers or teams that require dynamic off-site links. Using private networks of leased lines or frame-relay links, a company still has to maintain modem banks to provide connectivity to mobile workers, which has become more of a problem as the demand for mobile communications and remote access has increased.

Previous | Table of Contents | Next

The conventional response to corporate growth—adding another frame-relay link or modem bank—doesn't mesh well with today's dynamic business environments. The problem with leased lines and frame relay is that setting them up takes too long. And, even if the frame-relay circuits could be set up quickly enough, each WAN interface is expensive and requires attention, not only during setup but for ongoing maintenance. Although modems can be set up fairly quickly, they may not support the bandwidth needed, and they can involve higher management overhead in the form of remote user support. The management of the two systems also is not integrated.

Designing the Net

Because leased lines are dedicated to handling only traffic between two points, the number of lines in a simple network connecting all branch offices to the corporate headquarters grows linearly as the number of branch offices increases. But, this star network topology requires all traffic to pass through headquarters, which can be a single point-of-failure. If the connection to HQ goes down, communications between branch offices are cut as well.

One answer is to build in redundant links, forming a mesh including additional links between the branch offices, like that shown in Figure 2.1. But, that becomes an expensive solution, especially if the redundant links aren't used much. Another solution is to create what's called a *hub-and-spoke topology* (see Figure 2.2.), which makes it possible to maintain some local connectivity should one of the major connection points (a hub) go down.

FIGURE 2.2 A hub-and-spoke network.

FIGURE 2.3 A private network using a frame-relay net.

Nowadays, the situation has changed sufficiently to make further expansion of leased lines and larger modem banks both an expensive proposition and one requiring increased management and support resources. And, if flexible business arrangements are required with partners or temporary offices, or mobile teams of workers are needed, the delays associated with requesting and installing new leased lines

or frame-relay links become counter-productive if not downright unacceptable. What's required is a single solution that not only provides for the security of corporate traffic but also provides the flexibility of configuration and connectivity that today's businesses require. That solution is the Internet VPN.

Frame Relay Notes

Frame relay is a data-oriented network interface used to send bursts of data over a wide area network. As a packet-based technology, frame relay does not allocate bandwidth until real data is transmitted. Instead, frame relay defines virtual circuits in the network, known as permanent virtual circuits or *permanent virtual connections* (PVC). A PVC typically is defined between two corporate sites. Effectively, a PVC sets up a logical network connection between the sites over the shared frame-relay network. Unfortunately, you have to pay a monthly rental fee for each PVC you need to connect your sites, regardless of how much you use them. When you lease a PVC from a frame-relay provider, part of the agreement is a *Committed Information Rate* (CIR). This CIR sets the minimum bandwidth the provider guarantees will be available for your traffic 24 hours a day, 7 days a week. A CIR is not tied in any way to the speed of your physical connection; you could have a T1 connection, but pay for a 64-Kbps CIR.

What Is an Internet VPN?

Rather than depend on dedicated leased lines or frame relay's PVCs, an Internet-based VPN uses the open, distributed infrastructure of the Internet to transmit data between corporate sites. In essence, companies using an Internet VPN set up connections to the local connection points, called *Points-of-Presence* (POPs), of their *Internet Service Provider* (ISP) and let the ISP ensure that the data is transmitted to the appropriate destinations via the Internet, leaving the rest of the connectivity details to the ISP's network and the Internet infrastructure (see Figure 2.4).

The link created to support a given communications session between sites is dynamically formed, reducing the load on the network; permanent links aren't part of the Internet VPN's structure. In other words, the bandwidth required for a session isn't allocated until it's required and is freed up for other uses when a session is finished. In many ways, this aspect resembles the properties of a frame-relay network, but it's extended to other types of connections on the Internet.

FIGURE 2.4 An Internet VPN.

Because the Internet is a public network with open transmission of most data, Internet VPNs include the provision for encrypting data passed between VPN sites, which protects the data against eavesdropping and tampering by unauthorized parties.

As an added advantage, an Internet VPN also supports secure connectivity for mobile workers by virtue of the numerous dial-in connections that ISPs typically offer clients at their POPs.

Why Use an Internet VPN?

Whether you're building a VPN from scratch or converting your traditional VPN to one using the Internet, a number of benefits arise from the use of Internet-based VPNs. These benefits are direct and indirect cost savings, flexibility, and scalability.

Virtual Circuit or Tunnel?

Technically speaking, virtual circuits are restricted to a single type of transmission medium-frame-relay virtual circuits are one example. But, we are, in effect, creating virtual circuits between sites using the Internet for a VPN, so what's the difference? Because the Internet embraces a number of transmission media, an Internet VPN cannot rely on the mechanisms built into just one medium to form a virtual circuit but must depend on other protocols within the TCP/IP suite to form these virtual circuits.

The way that Internet VPNs create these virtual circuits is to encapsulate data packets within special IP packets for transmission on the Internet, enabling them to be transmitted on any medium that supports IP. To avoid any confusion with the media-dependent virtual circuits, the paths that the encapsulated packets follow in Internet VPNs are called *tunnels*, not virtual circuits.

Cost Savings

First and foremost are the cost savings of Internet VPNs when compared to traditional VPNs. A traditional VPN built using leased T1 (1.5 Mbps) links and T3 (45 Mbps) links has to deal with tariffs structured to include an installation fee, a monthly fixed cost, and a mileage charge. For example, a T3 line has an average fixed charge (without the mileage charge) in the range of $25,000 to $27,000 per month; the mileage pricing is around $60 to $65 per month, per mile. For a T1 line, the average fixed charge is $3,400 to $3,800 per month, with a mileage charge of $4 to $6 per month, per mile. For a leased line between New York and Chicago, a T1 would cost about $8,000 per month.

The costs associated with frame-relay networks differ from those for leased lines; frame-relay networks are usually less expensive than dedicated leased lines, but they also require fees for the Permanent Virtual Circuits that the provider allocates between each of your sites. A typical T1 connection to a frame-relay net would cost around $2,000 per month, with an additional cost of $1,400 per month for each PVC. Frame-relay fees do not include a charge for distance.

Internet Service Providers offer digital connections in a number of bandwidths: 56 Kbps, T1, fractional T1, burstable T1, T3, fractional T3, and burstable T3. Leased line prices from ISPs, which are not the same as an RBOC leased line because it only travels to the ISP's local POP, include a one-time installation fee and a monthly fixed fee, with no mileage charges. A dedicated T1 Internet circuit lists for around $2,400 per month; a full T3 circuit costs about $55,000 per month.

Previous | Table of Contents | Next

Leased Internet lines offer another cost advantage because many providers offer prices that are tiered according to usage. With Local Exchange Carriers [LECs], you pay the same fee for a fixed–bandwidth leased line, regardless of how much of the bandwidth you use and how often you use it. For businesses that require the use of a full T1 or T3 only during busy times of the day but don't need the full bandwidth the majority of the time, ISP services such as burstable T1 are an excellent option. Burstable T1 provides on–demand bandwidth with flexible pricing. For example, a customer who signs up for a full T1 but whose traffic averages 512 Kbps of usage on the T1 circuit will pay less than a T1 customer whose average monthly traffic is 768 Kbps if burstable T1 rates are used.

Eliminating long–distance charges is another cost savings resulting from Internet VPNs. Rather than require mobile employees or off–site teams to dial–in via long–distance lines to the corporate modem bank, a company's VPN enables them to place local calls to the ISP's POP in order to connect to the corporate network.

It's also conceivable that your costs can be reduced by outsourcing the entire VPN operation (aside from setting security rights for your employees) to the service provider. Some of the providers we discuss in Chapter 9 include full technical support, help–desk services, and security audits, which can reduce your own internal support requirements.

Some Detailed Cost Comparisons

It's often been written that the cost savings alone makes it worthwhile to adopt Internet VPNs in your business. Although it's impossible to offer enough details to cover all possible network configurations, this section includes three different network scenarios to show how costs differ between private networks using leased lines, the Internet, and remote–access–only. One scenario is aimed at a small company of three offices; one focuses on a large company with four regional/main offices and six branch offices; and the last covers a company interested in providing only remote access for its mobile workers.

In all cases, we've simplified the calculations somewhat by not including the charges for a local loop, which each site would need, and we've not included any support personnel costs. Each of these calculations is an approximation of the costs; your mileage may vary...

SCENARIO 1

This scenario (see Figure 2.5) is the simplest of the group, consisting of three offices located on the East Coast—Boston, New York City, and Washington D.C.—that want to have a full–time virtual network between them. They're running only a single T1 line between each office in the first part of this scenario.

Capital outlays for equipment and installation at each site include $2,000 per router, $1,000 for a CSU/DSU, and $300 for installation of the T1. The center link in the network (New York City) has to install two CSU/DSUs and two routers. The resulting setup cost is therefore $13,200. The T1 fees were figured as an average of late 1997 fees (i.e., $3,600 per month plus $5/mile/month). (See Table 2.1.)

For a network setup using an Internet VPN, the router and CSU/DSU costs are assumed to be the same as for the T1 case, but the initial installation costs are higher (i.e., $3,000 per site, adding up to a setup cost of $18,000). The Internet access fee for a T1–speed link to the ISP was assumed to be $1,900 per site.

FIGURE 2.5 Map of regional three–office network.

Although the T1 lines are less expensive to install than the Internet VPN, running a simple trunk, or bus, of T1 lines between the three sites costs almost three times as much per month. Given the preceding situation, MegaGlobal Corp. would recoup its expenditures for the Internet VPN in less than one month of operation. Obviously, if the company already had the capital equipment and switched from the leased lines to an Internet VPN, the time for recovering the costs would be even less.

The second part of this scenario has MegaGlobal Corp. create a mesh between all three cities for improved reliability (see Table 2.2). The assumptions are the same as before, but now each site has to install two CSU/DSUs and two routers for the leased lines (see Figure 2.5), which adds up to a capital outlay of $19,800. The Internet VPN setup costs remain the same as before.

TABLE 2.1 Monthly Costs for Single Leased–Line Networks versus Internet VPN

City	*Distance (mi.)*	*T1 Fees*	*Internet VPN Fee*
Boston–NYC	194	$4,570	$1,900
NYC–Washington DC	235	$4,775	$1,900
Total		$9,345	$3,800

TABLE 2.2 Monthly Costs for Leased–Line Mesh and Internet VPN

City	*Distance (mi.)*	*T1 Fees*	*Internet VPN Fee*
Boston–NYC	194	$4,570	$1,900
NYC–Washington DC	235	$4,775	$1,900
Boston–Washington DC	463	$5,915	$1,900
Total		$15,260	$5,700

SCENARIO 2

The second scenario describes company MegaGlobal Corp. with four major regional offices across the country—in San Francisco, Denver, Chicago, and New York City. MegaGlobal Corp. also has six additional branch offices in the United States, which it wants to connect to the regional offices. These offices are located in Los Angeles, Salt Lake City, Dallas, Minneapolis, Washington DC, and Boston.

For a leased–line network, MegaGlobal Corp. has chosen to use a hub–and–spoke model, with the four regional offices serving as hubs and the branch offices connecting to the closest hub on the spoke (see Figure 2.6). To improve reliability between the regional offices, two T1s are run between each hub; the branch offices have a single T1 each.

FIGURE 2.6 Map for national corporate network.

Capital outlays for equipment and installation at each site include $2,000 per router, $1,000 for a CSU/DSU, and $300 for installation of the T1. Because of the redundant lines, 24 CSU/DSUs and 24 routers are needed (assuming a separate device for each link). The resulting setup cost is therefore $79,200. The T1 fees were picked as an average of late 1997 fees (i.e., $3,600 per month plus $5/mile/month). (See Table 2.3.)

For a network using an Internet VPN, the router and CSU/DSU costs are assumed to be the same as for the T1 case, but the initial installation costs are higher (i.e., $3,000 per site, adding up to a setup cost of $60,000). The Internet access fee for a T1 speed link to the ISP was assumed to be $1,900 per site.

Previous | Table of Contents | Next

It's easy to see that the Internet VPN is a money saver after the first month of operation. Using single T1s between the hubs reduces the cost somewhat, to an initial setup cost of $59,400 and monthly fees of $60,655, but that doesn't significantly change the point at which the Internet VPN costs less than the T1 solution.

Even if lower–speed links, say 56 Kbps, were used for connecting the branch offices to the regional offices, the Internet solution would cost less.

SCENARIO 3

Because some products marketed as VPN products seek to replace dial–in remote access products with Internet access, this last scenario focuses on remote access only. In this case, MegaGlobal Corp. wants to support 100 remote users with dial–in access via the Internet. We are assuming that there will be 25 percent local calls and 75 percent long–distance calls into the office. We also assume that each worker using remote access averages one hour of connectivity per working day, for a total of 20 hours per month. Long–distance call charges average $10 per hour, which results in long–distance charges of $15,000 per month (0.75*2,000 hrs./month*$10/hr.). (See Table 2.4.)

TABLE 2.3 Monthly Costs for Leased–Line Netwotk and Internet VPN

City	*Distance (mi.)*	*T1 Fees*	*Internet VPN Fee*
SF–Denver	1,267	$13,535	$1,900
Denver–Chicago	1,023	$12,315	$1,900
Chicago–NYC	807	$11,235	$1,900
SF–LA	384	$5,520	$1,900
Denver–Salt lake	537	$6,285	$1,900
Denver–Dallas	794	$7,570	$1,900
Chicago–Minneapolis	410	$5,650	$1,900
NYC–DC	235	$4,775	$1,900
NYC–Boston	194	$4,570	$1,900
Total		$71,455	$17,100

Comparative monthly charges for the Internet VPN solution include the $20 per month ISP fees for each

user's dial–up account and the T1 access fee of $1,900 per month for the main site.

Required equipment or software that we haven't discussed before would be a terminal server for the remote–access case and security gateway software for the Internet VPN solution. Although MegaGlobal Corp. wants to support 100 remote users, we assume that it will provide only a fraction of that number of lines and a configured 10–port terminal server; at a cost of $550 per port, the terminal server would cost $5,500.

Capital outlays for the Internet VPN are the same as in previous scenarios, but only one router and CSU/DSU are needed because everyone is connecting to the main office. Thus, only one T1 line to the ISP has to be installed.

There's a wide variation in the cost of security software, as we'll see later in this book. At the low end, software bundled with Microsoft's Windows NT server is the most cost–effective. Assume that a suitable NT server and software license would run around $2,600 and do not factor in any additional client costs, assuming that each user already will have installed the appropriate version of Windows for their daily work. At the high end, the security gateway software for a router can cost around $15,000, with added costs for the client software (at $100 per user).

Thus, the capital outlay for the low–end Internet VPN solution would be $8,600, while the high–end solution costs $31,000 (T1 installation + router + CSU/DSU + security gateway software + 100 security clients). With a monthly savings of $11,100, the Internet VPN solution allows MegaGlobal Corp. to recoup its initial investment in one month for the low–end solution and in about three months for the high–end solution.

Are there occasions when the Internet VPN is not a cost–effective solution? A few. First, if a company has to use only a single leased–line between two locations that are relatively close, the fees for a T1 line can be less than the equivalent ISP installation for the Internet VPN. Second, if all of the sites are close to each other and form a small regional network, a set of leased lines can prove to be less costly. Third, if most of the remote users are local telecommuters that do not require long–distance calls, a modem bank will most likely be less expensive than ISP charges.

TABLE 2.4 Monthly Costs for Remote Access Via Direct Dial–in and Internet VPN

Direct Dial–in		***Internet VPN***	
Long–distance charges	$15,000	ISP dial–in accounts	$2,000
		T1 line	$1,900
Total	$15,000		$3,900

Using frame relay to form the private network also can bring the costs down, because no mileage fees are charged. But, with either solution, bear in mind that you'll still have to maintain a different infrastructure for dial–in access from mobile workers and telecommuters, which adds to the cost of capital equipment as well as network management and support. Internet VPNs still offer more flexibility and scalability than other alternatives.

Flexibility

With traditional VPNs, the other connections that serve smaller branch offices, telecommuters, and mobile works—xDSL, ISDN, and high–speed modems—have to be maintained with separate equipment (modem banks, for instance) that are not part of the setup for either leased lines or frame relay.

In an Internet–based VPN, not only can T1 and T3 lines be used between your offices and the ISP, but many other connection types can be used to connect smaller offices and mobile workers to the ISP and, therefore, to your VPN. The only restriction is the media that the ISP supports, and the number of supported media is constantly growing.

Previous | Table of Contents | Next

Because point–to–point links aren't a part of the Internet VPN, your company doesn't have to support the same media and speeds at each site, further reducing equipment and support costs. If your mobile workers are using 56–Kbps modems and telecommuters use ISDN to connect to the ISP and the Internet, the appropriate equipment is required only on their end, the client side. By the time their traffic makes its way to the corporate net, it's been aggregated with other corporate traffic and is being transmitted over the main connection that your corporate net maintains to the Internet, such as a T1 or T3 link (see Figure 2.7). The third scenario presented earlier is a good example of this.

Scalability

Because VPNs use the same media and underlying technologies as the Internet, they're able to offer businesses two dimensions of scalability that are difficult to achieve otherwise.

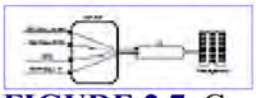


FIGURE 2.7 Consolidation of incoming traffic.

First, there's geographic scalability. With an Internet VPN, offices, teams, telecommuters, and mobile workers can become part of a VPN wherever the ISP offers a *Point–of–Presence* (POP). Most large ISPs have a significant number of POPs scattered throughout the United States and Canada, with many also offering POPs in Europe and Asia. This scalability also can be dynamic; a field office at a customer's site can be linked easily to a local POP within a matter of minutes (using a regular phone line and a modem, for instance) and just as easily removed from the VPN when the office closes up shop. Of course, higher bandwidth links may take longer to set up, but the task is often easier than installing a leased line on someone else's premises.

Second, there's bandwidth scalability. We've already mentioned that ISPs charge by usage, so fees for a little–used T1 are less than those for a highly used T1. But, ISPs can also quickly offer your choice of bandwidths according to the needs of your sites. Your home office may require a T1 or even a T3 connection, for instance, while your branch offices might be able to get by with a dial–up modem line or an ISDN line. And, if a branch office requires more bandwidth, it can upgrade from a plain phone line to a 56–Kbps or ISDN connection or from ISDN to a T1. Your network can grow as needed; since links aren't hard wired between each site, you don't have to upgrade the equipment at every site to support changes at one site.

Reduced Tech Support

VPNs also can reduce the demand for technical support resources. Much of this reduction stems from

standardization on one type of connection (IP) from mobile users to an ISP's POP and standardized security requirements. As mentioned earlier, outsourcing the VPN also can reduce your internal technical support requirements, because the service providers take over many of the support tasks for the network.

Reduced Equipment Requirements

Lastly, by offering a single solution for enterprise networking, dial–in access, and Internet access, Internet VPNs require less equipment. Rather than maintaining separate modem banks, terminal adapters, and remote access servers, a business can set up its *customer premises equipment* (CPE) for a single medium, such as a T3 line, with the rest of the connection types handled by the ISP. The IT department can reduce WAN connection setup and maintenance by replacing modem banks and multiple frame–relay circuits with a single wide area link that carries remote user, LAN–to–LAN, and Internet traffic at the same time.

Meeting Business Expectations

When it comes to integrating any new technology into a business network, a number of common concerns always have to be addressed. These concerns are standards, manageability, scalability, legacy integration, reliability, and performance.

Corporate managers and planners like to see that products and services comply with the common standards of the day, partly to ensure longevity of the products, but also, and perhaps more importantly, to ensure that products from different vendors will interoperate. Even though many companies still choose to go with a single vendor for their networking equipment, thus reducing the demand for vendor interoperability, these same companies still like to keep their options open should better– or lower–priced components become available.

As networks become more complicated and as the number of users increases, network managers find themselves between a rock and a hard place. Not only do they have to manage, monitor, and configure more network devices, but they usually have to perform these tasks with either a fixed or a reduced number of staff. It's rare to see the network staff grow as quickly as the network itself. Thus, adding any new components or services to the network has to fit into existing network management systems or, even better, the existing management tasks have to be simplified. And, considering the importance of security in VPNs, it's just as important that VPN security management fit nicely into a corporation's security plans.

Network managers must plan for growth as they review products and services for their networks. They don't want to be faced with replacing one product or technology with another in a few months or a year when the demand for a service increases. Using the same software, but on a faster server, for instance, is a scalability approach managers can deal with; scrapping both the software and the hardware isn't. Similarly, a vendor offering a series of hardware products that offer the same functionality, but support more users or faster bandwidth, often fits better into a network designer's plan than does a company offering only one solution.

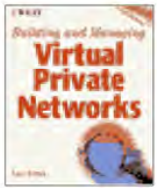
These days, few, if any, businesses have the luxury of starting their computing and networking infrastructure from scratch. There's a great deal of older data, systems, and networks that usually have to be supported in order for a company to continue operating. These legacy systems have to mesh with new

systems somehow. We'll see that some VPN solutions offer better choices for supporting multiple protocols, for instance, making them better suited for integration with legacy networks. But, other solutions may require conversion to a suite of protocols only, and such changes have to be taken into account as you plan.

Another factor of great importance to network managers is the reliability of the product or service. For VPNs, reliability concerns focus on two different components—the hardware (and associated software) and the communications services (i.e., the Internet). Using standard components in the hardware—microprocessors, proven interface cards, and so on—is important, as is the maintainability of the hardware. Modular construction of a device is a plus, as is the capability to maintain some semblance of continued operation while the device is being maintained. Concerns about the reliability of the Internet as a data transmission channel have been frequently raised, but many ISPs have been working on ways to guarantee better reliability.

A related concern, especially for the Internet, is that of guaranteed performance. As a network manager, you want to ensure that data traffic goes through as expected, with the right amount of bandwidth assigned to high–priority and low–priority traffic. Delays should be minimized. Service *Level Agreements* (SLAs) with your network provider are a must; even now, SLAs are still evolving, as customers demand more services and assurances and providers roll out new features for improving their services. We'll see in later chapters that many of the methods used to secure data traffic can be computationally intensive, making it necessary to plan for the possible deleterious impact cryptographic processing may have on normal network flows.

Previous | Table of Contents | Next



Building and Managing Virtual Private Networks
by Dave Kosiur
Wiley Computer Publishing, John Wiley & Sons, Inc.
ISBN: 0471295264 **Pub Date:** 09/01/98

Although there's a proven demand for Internet–based VPNs, the market is still in its relative infancy, as protocols and devices continue towards some semblance of standardization. A number of issues can impact both your design and deployment of an Internet VPN; we'll mention them briefly here and present more details in the following chapters of this book.

Security. An Internet VPN should be only one part of a company's security plan. Securing tunnels for private communications between corporate sites will do little if employee passwords are openly available or if other holes are in the security of your network. At the same time, VPN–related security management, of keys and user rights, for instance, have to be integrated into the rest of your company's security policies.

International operations also pose an added security problem. The export of advanced encryption software is restricted not only by the U.S. government but also by many other governments. Mobile workers using 128–bit encryption to secure transmissions in the United States are still largely restricted from using anything more than 56–bit encryption when traveling abroad.

Potential bottlenecks. Encryption and decryption can be computationally intensive processes that can lead to reduced throughput if your security gateway has insufficient computing horsepower. For high–bandwidth links, hardware–based encryption or at least a dedicated high–speed workstation running encryption software are likely solutions. Software–based encryption on shared hardware (a firewall or remote access server, for instance) can be sufficient for lower bandwidth connections, such as 56 Kbps or ISDN.

Packet encapsulation increases the size of the original packets, which may make them larger than the sizes normally based by routers and other network devices. In such cases, packets are fragmented, which can lead to poorer performance. One solution is to compress the original packets before encapsulation. VPNet Technologies offers this option in their products and the IPSEC Working Group is investigating ways to standardize this approach.

Interoperability. The current slate of protocols for tunneling and security are not interoperable. Yet, selection of a single protocol to meet all of your VPN needs is problematic, because protocols like PPTP and L2TP are better suited for client–initiated tunnels, while IPSec is best for LAN–to–LAN tunnels. The fact that most of the protocols are converging on IPSec for encryption improves interoperability, but you initially may find it necessary to use devices that support more than one tunnel protocol.

If your company has an existing WAN infrastructure, the costs of implementing a VPN will be lower if you can utilize much of the existing equipment. But, that may pose problems of interoperability with the new VPN equipment; for instance, some branch offices may not be able to upgrade their CPE to meet the requirements of a VPN.

IP address management. If a corporate VPN is designed as one network with some special routed

links called tunnels, full routing is possible between the parts of your network that are connected by tunnels, and you can use a single unified *Domain Name Service* (DNS) for resolving device names and IP addresses. This makes both reachability of hosts and routing more convenient and easier to manage.

But, the more common situation is one in which each part of the VPN is treated as a separate network, again with some tunneled and routed links. Unfortunately in this case, it's difficult to find a unified routing table, and the DNS also might be fairly fragmented, adding to the difficulty of managing the VPN.

Other management issues revolve around deciding which private IP addresses should stay private or be handled by NAT and how other DNS information is provided to parts of the VPN.

Reliability and performance. Because Internet–based VPNs depend on the Internet, they are subject to the same performance problems that Internet traffic experiences. One solution is not to use the public Internet for your VPN but to employ a service provider's private IP network, although even these networks have not yet reached the reliability of traditional networks.

Furthermore, Internet VPNs can incur reliability and performance problems due to congestion, dropped packets, and other factors, which could cause problems for real–time applications, such as telephony and videoconferencing. Also, the encapsulated IP headers found in tunnels can cause problems for some QoS schemes, preventing them from allocating the appropriate network resources.

Multiprotocol support. Even though many companies are switching to TCP/IP as the protocol suite of choice for their networks, other protocols are still important in legacy systems and networks. NetWare's IPX is one such example. Tunneling non–IP packets over IPSec is a problem, because IPSec is designed for encapsulating only IP packets. On the other hand, PPTP and L2TP include more multiprotocol support in their tunnels. If large enterprise VPNs are part of your design and neither PPTP nor L2TP can scale to your needs, you may have to include upgrading other services based on non–IP protocols to IP in order to create your VPN. Netware can now be run over IP instead of IPX, for example.

Integrated solutions. Although a single–source solution in one device may sound like a good solution, especially from security and network management viewpoints, bear in mind that a single integrated device also can become a single point of failure—if it goes down, you've lost all of your VPN capabilities.

Summary

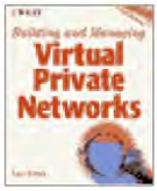
Private networks designed to link together a number of corporate sites have used a variety of technologies over the past 30 years. But, it's only been recently that alternatives to using dedicated leased lines, like frame relay, have seen more use. These newer technologies have enabled businesses to replace expensive leased lines with less expensive, dynamic links, or virtual circuits.

Internet VPNs go a step farther by offering businesses the opportunity to create these dynamic links over a variety of different transmission media, thus offering a single form of protected connectivity for both LANs at different sites and mobile workers. In addition to offering better flexibility and scalability, Internet VPNs can offer significant cost savings.

Being a relatively recent development, Internet VPNs still have some issues to deal with, such as

guaranteed performance and security, but these issues are being actively addressed by the commercial providers as well as standards–setting bodies like the IETF.

Previous | Table of Contents | Next



Building and Managing Virtual Private Networks
by Dave Kosiur
Wiley Computer Publishing, John Wiley & Sons, Inc.
ISBN: 0471295264 **Pub Date:** 09/01/98

CHAPTER 3
A Closer Look at Internet VPNs

Using the Internet to create Virtual Private Networks presents considerable advantages to corporate users, as we saw in the preceding chapter. To provide a better understanding of the unique structure and processes of Internet VPNs, this chapter presents an overview of the tunneling and security features of the Internet, the protocols involved, and the various types of equipment available for building VPNs. The full details of many components are covered in following chapters.

The Architecture of a VPN

Two fundamental components of the Internet make VPNs possible. First, the process known as tunneling enables the virtual part of a VPN; second, various security services keep the VPN data private.

Tunnels: The "Virtual" in VPN

In VPNs, "virtual" implies that the network is dynamic, with connections set up according to the organizational needs. Unlike the leased-line links used in traditional VPNs, Internet VPNs do not maintain permanent links between the endpoints that make up the corporate network. Instead, a connection is created between two sites when it's needed. When the connection is no longer needed, it's torn down, making the bandwidth and other network resources available for other uses.

Virtual also means that the logical structure of your network is formed only of your network devices, regardless of the physical structure of the underlying network (the Internet, in this case). Devices such as routers or switches that are part of the ISP's network are hidden from the devices and users of your virtual network. Thus the connections making up your VPN do not have the same physical characteristics as the hard-wired connections used on your local area network (LAN), for instance. Hiding the ISP and Internet infrastructure from your VPN applications is made possible by a concept called *tunneling*.

Tunnels are used for other services on the Internet besides VPNs, such as multicasting and mobile IP. Tunneling creates a special connection between two endpoints. To create a tunnel, the source end encapsulates its packets in IP packets for transit across the Internet. For VPNs, the encapsulation may include encrypting the original packet and adding a new IP header to the packet (see Figure 3.1). At the receiving end, the gateway removes the IP header and decrypts the packet if necessary, forwarding the original packet to its destination (see Figure 3.2).

Tunneling allows streams of data and associated user information to be transmitted over a shared network within a virtual *pipe*. This pipe makes the routed network totally transparent to users.

Ordinarily, tunnels are defined as one of two types—permanent or temporary. But, *static tunnels*, as the first kind are often called, are of little use for VPNs, because they will tie up bandwidth even if it's not being used. Temporary, or dynamic, tunnels are much more interesting and useful for VPNs, because they can be set up as needed and then torn down after they're no longer needed (e.g., when a communications session is finished). Dynamic tunnels, therefore, don't require constant reservation of bandwidth. Because many ISPs offer connections priced according to the average bandwidth used on a connection, dynamic tunnels can reduce the bandwidth utilization and lead to lower costs.

Tunnels can consist of two types of endpoints, either an individual computer or a LAN with a security gateway, which might be a router or firewall, for instance. Only two combinations of these endpoints are usually considered in designing VPNs, however. In the first case, LAN-to-LAN tunneling, a security gateway at each endpoint serves as the interface between the tunnel and the private LAN (see Figure 3.3). In such cases, users on either LAN can use the tunnel transparently to communicate with each other.

FIGURE 3.1 A packet prepared for tunneling.

The second case, that of client-to-LAN tunnels, is the type usually set up for a mobile user who wants to connect to the corporate LAN. The client (i.e., the mobile user) initiates the creation of the tunnel on his end in order to exchange traffic with the corporate network. To do so, he runs special client software on his computer to communicate with the gateway protecting the destination LAN.

Security Services: The "Private" in VPN

Equally important to a VPN's use, if not more so, is the issue of privacy or security. In its most basic use, the "private" in VPN means that a tunnel between two users on a VPN appears as a private link, even if it's running over shared media. But, for business use, especially for LAN-to-LAN links, *private* has to mean more than that; it has to mean security, that is, freedom from prying eyes and tampering.

FIGURE 3.2 Schematic of a tunnel.

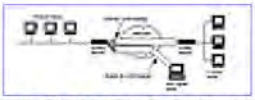


FIGURE 3.3 LAN and client VPN tunnels.

Today's Internet is a large cloud of interconnected networks, with most of its traffic being transmitted as open, or unencrypted, data. A prime requirement, then, for creating an Internet-based VPN is security.

VPNs need to provide four critical functions to ensure security for your data. These functions are as follows:

Authentication. Ensuring that the data is coming from the source from which it claims to come.

Access control. Restricting unauthorized users from gaining admission to your network.

Confidentiality. Preventing anyone from reading or copying your data as it travels across the Internet.

Data integrity. Ensuring that no one tampers with data as it travels across the Internet.

Although tunnels can ease the transmission of your data across the Internet, authenticating users and maintaining the integrity of your data depends on cryptographic procedures, such as digital signatures and encryption. These procedures use shared secrets called *keys,* which have to be managed and distributed with care, further adding to the management tasks of a VPN (see Chapter 4, "Security: Threats and Solutions," for more details).

Although security services can be applied at different layers of the communications stack, such as the Application layer, Session layer, and Network layer, our focus while describing Internet VPNs will be the services for authentication, encryption, and data integrity that are offered at layers 2 and 3 of the OSI model—that is, the Data-Link and Network layers. Deploying security services at the lower OSI layers makes much of the security services transparent to the user.

Previous | Table of Contents | Next

VPNs and IP Addresses

In addition to hiding the underlying structure of the Internet between two endpoints, tunnels make preventing address conflicts that can arise when connecting two LANs easier.

Although the 4.3 billion addresses provided by the 32-bit IP address in IPv4 may seem adequate for most internetworking, companies may encounter a shortage of addresses if they want to communicate openly over the Internet. But, if a company wants to create its own private internet that does not connect to the global Internet, that company can use all of the 32-bit address range for its own network devices and computers. (Recommended practices reduce the available address space somewhat; see Chapter 14, "IP Address Management," for more details.) What happens when two previously isolated private IP networks want to connect and communicate with each other over the Internet? Even if we assume that the two networks are not using the same addresses, their private addresses are likely to conflict with other addresses used on the public Internet, which will cause routing problems. But, encapsulating packets to hide the privately assigned addresses relieves this problem. Part of the packet encapsulation process performed by a tunnel endpoint includes adding a new address to the packet; this address is the one corresponding to the other endpoint of the tunnel. Any forwarding of the encapsulated packet through the tunnel that must be done on the Internet is done using this address, not the address of the actual destination. Thus, only the addresses of the tunnel endpoints have to be IP addresses that are legitimate within the Internet IP address space, regardless of how many users with privately assigned IP addresses send data through the tunnel.

But, implementation of security at these levels can take two forms, which affect the individual's responsibility for securing his own data. Security can be implemented either for end-to-end communications (i.e., between two computers) or between other network components, such as firewalls or routers. This last case is often referred to as node-to-node security (see Figure 3.4).

FIGURE 3.4 End-to-end versus node-to-node security.

Using security on a node-to-node basis can make the security services more transparent to the end-users and relieve them of some of the heavy-duty computational requirements, such as for encryption. But, node-to-node security expects—in fact, requires—that the networks behind the node must be trusted networks (i.e., secure against other attacks that unauthorized users might try). End-to-end security, because it involves each host, the sender and the receiver, directly, is inherently more sound than node-to-node security. End-to-end security comes with its own disadvantages; namely, it increases complexity for the end-user, and it can be more challenging to manage.

Now let's take a look at the way in which different network components fit together in an Internet VPN.

The Protocols behind Internet VPNs

Two major classes of protocols make VPNs possible on the Internet. First, there are the protocols that define how packets are encapsulated and tunnels formed, as well as how the packets are secured. Second, because the security protocols often involve the exchange of secrets between senders and receivers on the VPN, protocols are needed for handling the management of these secrets (i.e., cryptographic keys) and other authentication methods.

Tunneling and Security Protocols

Four protocols were originally suggested as VPN solutions. Three are designed to work at Layer2, the Link layer: the *Layer2 Forwarding* (L2F) protocol, the *Point-to-Point Tunneling Protocol* (PPTP), and the *Layer2 Tunneling Protocol* (L2TP). In an effort to improve interoperability and security while decreasing the proliferation of redundant, or near-redundant, protocols, the IETF is shepherding work on L2TP, which combines many of the features of L2F and PPTP. Because it's likely that L2F will soon be supplanted by L2TP, we'll focus on PPTP and L2TP as Layer2 solutions. The only VPN protocol for Layer3 is IPSec, which has been developed by the IETF over the past few years. (Socks is another protocol that can be used for VPNs, and it's handled at the application layer; we cover it briefly later) all the protocols are highlighted in Table 3.1.)

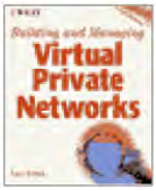
The details of each of these protocols are covered in later chapters; in the meantime, here's a quick run-down on their features:

- PPTP is a point-to-point tunneling mechanism originally created to support packet tunneling in Ascend's remote access server hardware and Microsoft's Windows NT software.
- The backers of PPTP combined efforts with Cisco and its L2F protocol to produce a hybrid Layer2 tunneling protocol called Layer2 Tunneling Protocol.
- IPSec is a standard created to add security to TCP/IP networking; it is a collection of security measures that address data privacy, integrity, authentication, and key management, in addition to tunneling.

All three VPN technology types that we consider here—PPTP, L2TP, and IPSec—support tunneling. PPTP and L2TP are strictly tunneling protocols. The tunneling mechanisms differ on what's done to the data (for instance, encryption and authentication), the headers that describe the data transmission and packet handling, and the OSI layer at which they operate.

Neither PPTP nor L2TP include encryption or key-management mechanisms in their published specifications. The current L2TP draft standard recommends that IPSec be used for encryption and key management in IP environments; future drafts of the PPTP standard may do the same. Although IPSec provides packet-by-packet encryption and authentication, it does not specifically cover management of the cryptography keys that would have to be exchanged (see the next section). Another working group in the IETF has been creating standards for key management in conjunction with IPSec; the protocol proposed for key management is called *ISAKMP/Oakley* (ISAKMP stands for *Internet Security Association and Key Management Protocol*), which is covered in detail in Chapter 5, "Using IPSec to Build a VPN."

Previous | Table of Contents | Next



Building and Managing Virtual Private Networks
by Dave Kosiur
Wiley Computer Publishing, John Wiley & Sons, Inc.
ISBN: 0471295264 **Pub Date:** 09/01/98

TABLE 3.1 VPN Protocol Comparisons

Technology	*Strengths*	*Weaknesses*	*Place in Network*
IPSec	+Standards track protocol. +Works independently of higher level applications. +Built as part of IPv6. +Allows for network address hiding without Network Address Translation. +Will accommodate developing cryptographic techniques.	+No user management. +Little production interoperability among vendors. +Little desktop support.	+Best at edge of network domain to be secured or on individual LAN segments. +Software best on the user's computer for vendor proprietary solutions for dial-up remote access.
PPTP	+Runs from Windows NT, Windows95, and Windows 98 platforms. +Accommodates end-to-end and node-to-node tunneling. +Popular value-added feature for remote access. +Uses existing Windows user domains for authentication. +Provides multiprotocol capability. +Uses RSA RC-4 encryption.	+Does not provide data encryption from remote-access servers. +Largely proprietary, requiring a Windows NT server to terminate tunnels. +Uses only RSA RC-4 encryption.	+Best in remote-access servers for proxy tunneling. +Can be used between remote offices that have Windows NT servers running RRAS. +Can be used on Windows95 desktops or Windows NT workstations.
L2F	+Enables multiprotocol tunneling. +Supported by many vendors.	+No encryption. +Weak user authentication. +No tunnel flow control.	+Best for remote access at POP.
L2TP	+Combines PPTP and L2F. +Needs only a packet based network to run over X.25 and frame relay. +Uses IPSec for encryption.	+Not yet implemented in many products. +Final mile unsecured.	+Best for remote access at POP.

Socks v5	+Contains application- level security for tighter control over access to applications. +Provides desktop-to-server authentication and encryption.	+Socks server can be resource-intensive, hampering scalability. +Can be difficult to manage outside users, such as trading and development partners. +Requires modification required.	+Best on the edge of the network behind a firewall. +Can be used on internal networks for user control. +Anywhere strong user authentication is to applications.

IPSec is often considered the best VPN solution for IP environments, because it includes strong security measures, notably encryption, authentication, and key management, in its standards set. Because IPSec is designed to handle only IP packets, PPTP and L2TP are more suitable for use in multiprotocol non-IP environments, such as NETBEUI, IPX, and AppleTalk.

Another protocol, SOCKS, is occasionally mentioned as a protocol for forming VPNs. SOCKS is designed to permit a datastream to cross a firewall based on user authentication rather than on the characteristics of the IP packets, such as a destination's UDP port number, which is the way firewalls usually work. SOCKS operates at the TCP layer and above, which makes establishing application-specific tunnels easier. For more details, check out Chapter 6, "Using PPTP to Build a VPN."

Management Protocols

Maintaining the access rights of your users and the security information, such as cryptographic keys, that relates to them is a crucial management issue in VPNs. Two different sets of protocols are currently used according to the type of VPN that's being maintained. For dial-in or client-to-LAN VPNs—using PPTP and L2TP tunnels, for instance—a protocol called RADIUS can be used for authentication and accounting. For LAN-to-LAN VPNs, much of the management of IPSec focuses on key management using the ISAKMP/Oakley protocol. (Many of the details of these protocols are covered in Chapters 4 and 5.)

The most popular tool for managing authentication and accounting for remote access has been the *Remote Authentication Dial-In User Service* (RADIUS), and it's the preferred protocol for use with dial-in tunneling, such as in PPTP and L2F.

RADIUS supports the authentication and accounting with a database that maintains access profiles for all trusted users. The information in each user's profiles includes passwords (authentication), access privileges (authorization), and network usage (accounting). The network access equipment interacts with the RADIUS server securely, transparently, and automatically. When a user attempts to log on remotely, the network access switch queries the RADIUS server to obtain that user's profile for authentication and authorization. A proxy RADIUS capability lets the RADIUS server at a service provider access an organization's RADIUS server to obtain any necessary user information, which is necessary to secure Internet-based VPNs.

As mentioned before, many authentication and encryption methods used in VPNs require the determination and distribution of keys. For small systems, manual distribution of keys, such as face-to-face, over a secure phone conversation, or via a courier, will suffice, but more automated systems are needed for larger VPNs. Although no standard is required for manual key management, some standardization is required for automated systems, partly because all network access equipment must regularly and automatically interact with the key-management system. In the ISAKMP/Oakley protocol that's being standardized by the IETF, ISAKMP defines the method for distributing keys, while the Oakley part specifies how keys are determined. (See Chapter 5, "Using IPSec to Build a VPN," for more details.)

VPN Building Blocks

If you take a look at Figure 3.5, you'll see that there are four main components of an Internet VPN: the Internet, security gateways, security policy servers, and certificate authorities. Not all of these components are defined or used in every current VPN product, but for the moment, we'll describe the most general case to show what the components are and how they fit together.

The Internet

The Internet provides the fundamental plumbing for your VPN. Although a great deal of the work of an Internet VPN takes place behind the scenes, it's worthwhile to understand how the Internet works. This knowledge will help you understand not only what your ISP can provide, but also why certain techniques are required for the success of Internet VPNs.

A number of components and players are along the path of a message you send, for example. Different types of Internet Service Providers (ISPs) are available, ranging from small local ISPs to regional ISPs and national or supranational ISPs, all arranged in tiers according to their capabilities.

Tier One providers such as FiberNet, AT&ampT, IBM, GTE Internetworking, and PSInet own and operate private national networks with extensive national backbones. These independent networks meet and interconnect at the Internet *Network Access Points* (NAPs). Through peering agreements between these private companies, the orderly exchange of digital traffic is facilitated between the various networks. In other words, the networks interconnect and exchange traffic at the NAPs to form what is essentially the Internet.

There are six Internet industry-recognized NAPs in North America. They are the Chicago AADS NAP, which is managed by Ameritech; the Sprint NAP, which is managed by Sprint; the MAE East NAP, which is managed by MFS; the MAE West NAP, which is managed by MFS; the PAC Bell NAP, which

is managed by Pacific Bell; and the CIX NAP, which is managed by The Commercial Internet Exchange.

A Tier Two provider is a company that buys its Internet connectivity from one of the Tier One providers and then provides residential dial-up access or World Wide Web site hosting or resells the bandwidth. It is important to note that none of the Internet NAPs provide Internet connectivity to the general public or to business and industry. The NAPs are only points for the orderly exchange of traffic between those organizations that maintain extensive national backbones. A NAP is not a point at which businesses or individuals can purchase Internet access. Additionally, connections to the Internet NAPs are made at a minimum of DS-3 speed (45 Mbps). The purpose of the Internet NAPs is to facilitate the orderly exchange of traffic from one network to another, not to sell Internet connectivity.

To become an industry-recognized NAP requires a substantial investment in Layer2 switching equipment and POP facilities. Typically, these facilities have redundant fiber optic cable paths to multiple carriers and can support circuit sizes up to and including OC-48 (2.4 Gbps).

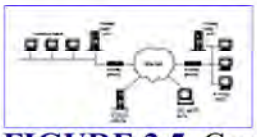


FIGURE 3.5 Components of an Internet VPN.

As an example of where your data is likely to travel, let's assume that you used a modem to dial your ISP's local Point-of-Presence to connect to the Internet and onto your corporate VPN (see Figure 3.6). The data travels from your laptop to the local POP and then on to the regional Internet network and probably over a few more POPs to the proper NAP before it's routed to another POP closer to the intended destination. There are two significant reasons why this all works: First, the different ISPs running the networks that make up the Internet cooperate with each other; second, the addressing features found in the IP protocol suite help tie all the networks together.

Whether you're an individual working at home or on the road and dialing into the Internet or a business with a full-time link to the Internet, the ISP's POP is an important cog in your use of the Internet. The POP is where the ISP handles the different types of media that its customers use for Internet access and forwards all the customer traffic to its backbone network, which connects to the rest of the Internet at some point (see Figure 3.7).

Some POPs contain different equipment for each transmission media they support, such as a modem bank for dial-in sessions and CSU/DSUs for frame relay and DDS; other ISPs have opted to leave support for the different media to the public network, instead running a leased line to their POPs. In addition to handling different media for customer traffic, the POP includes routers and/or IP switches to connect the POP's local LAN to the rest of the ISP's network as well as network management consoles. In some cases, the POP includes servers for hosting mail, news, Web sites, and RADIUS authentication servers for ISP's customers.

ISP or NSP?

Internet-related service providers are occasionally divided into two classes: Internet Service Providers (ISPs) and Network Service Providers (NSPs). Although ISPs offer Internet access only, NSPs offer dedicated IP bandwidth on private backbones, in addition to Internet access. UUNET and AT&ampT Worldnet are two examples of NSPs. Unless it's absolutely necessary to distinguish ISPs from NSPs, we use one term, ISP, to refer to both.

FIGURE 3.6 Communicating via ISPs, POPs, and NAPs.

Security Gateways

Take another look at Figure 3.5. Aside from the Internet cloud, which was just described, at least in part, the most significant components are those involving security. We not only have security gateways, but policy servers and certificate authority servers.

Security gateways sit between public and private networks, preventing unauthorized intrusions into the private network. They also may provide tunneling capabilities and encrypt private data before it's transmitted on the public network.

Previous | Table of Contents | Next

In general, a security gateway for a VPN fits into one of the following categories: routers, firewalls, integrated VPN hardware, and VPN software.

Because routers have to examine and process every packet that leaves the LAN, it seems only natural to include packet encryption on routers. Vendors of router-based VPN services usually offer two types of products, either add-on software or an additional circuit board with a coprocessor-based encryption engine. The latter product is best for situations that require greater throughput. If you're already using the vendor's routers, then adding encryption support to these routers can keep the upgrade costs of your VPN low. But adding the encryption tasks to the same box as your router increases your risks; if the router goes down, so does your VPN. For more details on the use of routers in VPNs, see Chapter 10, "Firewalls and Routers."

FIGURE 3.7 Schematic of a typical ISP POP.

Many firewall vendors include a tunnel capability in their products. Like routers, firewalls have to process all IP traffic—in this case, to pass traffic based on the filters defined for the firewall. Because of all the processing performed by firewalls, they're ill-suited for tunneling on large networks with a great deal of traffic. Combining tunneling and encryption with firewalls is probably best used only on small networks with low volumes of traffic. Also, like routers, firewalls can be a single point of failure for your VPN. For more details on firewall-based VPNs, see Chapter 10.

Another possible VPN solution is to use special hardware that's designed for the task of tunneling and encryption. These devices usually operate as encrypting bridges that are typically placed between the network's routers and WAN links. Although most of these hardware devices are designed for LAN-to-LAN configurations, some products also support client-to-LAN tunneling. See Chapter 11, "VPN Hardware," for more details.

IP Addresses and the Internet

The routers that connect all networks that make up an IP internetwork, such as the Internet, forward traffic based on the IP address of the destination network. To help with the assignment of a large number of addresses, IP addresses are divided into three major classes: A, B, and C. A fourth class, D, is reserved for special uses such as multicasting. See Table 3.2. Each address consists of four octets, or sets of eight binary digits, separated by decimals. The first octet determines which class the IP address is in. Class A addresses use the last three octets to specify IP nodes; Class B addresses use the last two octets for this purpose; and Class C addresses use the last octet.

TABLE 3.2 Address Classes and Numbers of Nets and Hosts

Class	*Network ID*	*# Unique Networks*	*Host Address ID*	*# Unique Hosts*
A	7 bits	128	24 bits	16,777,216
C	14 bits	>16,000	16 bits	65,536
C	21 bits	>2 million	8 bits	256

Class A network addresses are the most desirable, because they are large enough to serve the needs of any size enterprise. But, because fewer than 128 Class A networks can exist in the entire Internet, they are very scarce, and no more Class As are being allocated. Only those organizations that were early users of the Internet (e.g., Xerox Corp., Stanford U., and BBN) are in possession of Class A network addresses.

The more than 16,000 possible Class B networks also have become scarce and are now difficult to obtain. A large supply (more than 2 million) of Class C network addresses exist, so they are still plentiful. The major problem is that for most organizations, a Class C network is too small (only 256 unique host IDs). Even a Class B network is not large enough for an enterprise with more than a thousand LANs.

Lastly, software VPN systems are often good low-cost choices for environments that are relatively small and don't have to process a lot of traffic. These solutions can run on existing servers and share resources with them, and they serve as a good starting point for getting familiar with VPNs. Many of these systems are ideal for client-to-LAN connections and are covered in detail in Chapter 12, "VPN Software."

Other Security Components

Another important component of a VPN is the security policy server. This server maintains the access control lists and other user-related information that the security gateway uses to determine which traffic is authorized. For some systems, such as those using PPTP, access can be controlled via a RADIUS server; when IPSec is used, the server is responsible for managing the shared keys for each session.

Companies can choose to maintain their own database of digital certificates for users by setting up a corporate certificate server. For small groups of users, verification of shared keys may require checking with a third-party that maintains the digital certificates associated with shared cryptographic keys; these third parties are called *certificate authorities* (CAs). If a corporate VPN grows into an extranet, then an outside certificate authority also may have to be used to verify users from your business partners.

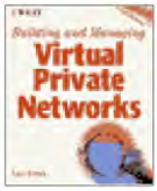
See Chapter 4, “Security: Threats and Solutions,” for more details on cryptographic keys, digital certificates, and certificate authorities.

Summary

The Internet VPN depends on creating media-independent tunnels across the Internet to transmits packets between sites. Because the Internet is an open communications environment that can be subject to unauthorized interception and access, other measures have to be taken to keep corporate data on a VPN private. Encryption is thus an integral part of VPNs on the Internet.

At the moment, IPSec is the most complete protocol for VPNs, especially when coupled with ISAKMP/Oakley for managing cryptographic keys. Other protocols that have been proposed for VPNs, namely PPTP and L2TP, provide better support for multiprotocol networks but will probably still require deployment of IPSec’s security features.

Previous | Table of Contents | Next



Building and Managing Virtual Private Networks
by Dave Kosiur
Wiley Computer Publishing, John Wiley & Sons, Inc.
ISBN: 0471295264 **Pub Date:** 09/01/98

PART II
Securing an Internet VPN

A main component of Internet-based VPNs is security. The three major protocols proposed for VPNs—IPSec, PPTP, and L2TP—each provide differing degrees of security for your data and ease of deployment. Standardization efforts will make IPSec and L2TP the preferred protocols over the next few years.

When designing your VPN, you should take into account how the strengths and weaknesses of each protocol mesh with the business and data needs of your network. PPTP and L2TP are aimed more at remote access VPNs, while IPSec currently works best for connecting LANs.

CHAPTER 4
Security: Threats and Solutions

One of the primary concerns of any corporation is protecting its data; fortunes can be made and lost, or reputations ruined, if information ends up in the wrong hands. Securing data against illegal access and alteration is even more of an issue on networks; transmitting data between computers or between LANs can make the data more vulnerable to snooping and interception than if it had remained on a single computer.

Many of the potential threats to transmitting data over today's networks are fairly well-known, and security experts know how to counter them. This chapter starts with a brief review of the common security threats in networked environments and then moves on to discuss the various cryptographic methods that enable you to protect your data on networks. With this background, we'll move on to the protocols and systems that implement these security solutions in the following chapters.

You should keep in mind that there's more to corporate security than what's covered in this chapter. A proper security framework for an organization includes seven different elements: authentication, confidentiality, integrity, authorization, nonrepudiation, administration, and audit trails (see Figure 4.1). The first three elements are the focus of this chapter, and we'll cover administration and auditing later in this book when we get to the section on VPN management. We'll leave it to other books, such as *Internet Security for Business,* by Terry Bernstein et al. (John Wiley & Sons, Inc., 1996) and *Computer Security Handbook,* edited by Arthur E. Hutt et al. (John Wiley & Sons, Inc., 1995), to cover some of the other details, such as configuring firewalls and setting up corporate security policies. Networking security is

just one part—albeit an important part these days—of corporate security and should fit in with your corporate security policies.

FIGURE 4.1 The components of a secure system.

Because the TCP/IP protocols were not designed with built-in provisions for security, many different security systems have been developed for applications and traffic running on the Internet. The software that's responsible for preparing data for transmission on a network offers a number of possibilities in which authentication and encryption can be applied. You could match each application of authentication and encryption to a specific protocol layer, using the 7-layer OSI Reference Model, for instance. But, for our purposes, it's sufficient to think of everything occurring in one of three layers: the application software, the network/transport stack, and the data link device and driver (see Figure 4.2). Some of the current-day cryptographic protocols for applications include *Secure MIME* (S/MIME) and *Pretty Good Privacy* (PGP) for e-mail and *Secure Sockets Layer* (SSL/TSL) and *Secure HTTP* (SHTTP) for Web applications. But, of greatest importance to the construction of VPNs is authentication and encryption at the Network and Data Link layers.

As we go through this chapter, recall that these methods can be applied to many different kinds of data and applications, but implementation in the Network layer is of primary importance to VPNs.

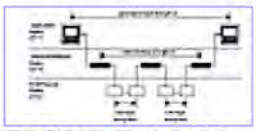


FIGURE 4.2 Network-layer versus Link-layer encryption.

Security Threats on Networks

In networked environments, the security of your data and communications depends on three things: authentication, confidentiality, and data integrity. Authentication means that the person with whom you're communicating really is that person; it's a step beyond identification because you're also verifying the identification. Maintaining the confidentiality of your communications is ensuring that no one can eavesdrop on your communications—that is, no one can read your data even if they intercept it. Lastly, guaranteeing the integrity of your data means that the data has not been altered in any way during transmission.

Unfortunately, as originally designed, the TCP/IP protocols and the networks built using these protocols, like the Internet, make it difficult to ensure that these three security features can be routinely provided. In the absence of proper security measures, data transmissions on IP networks can be subjected to a variety of threats. We'll review the more common types—spoofing, session hijacking, sniffing, and the man-in-the-middle attack—before moving on to the solutions that can defeat these attacks.

Spoofing

Like other networks, IP networks use a numeric address for each device attached to the network. The

address of the source and intended recipient is included in each data packet transmitted on an IP network. *Spoofing* takes advantage of the fact that an attacker can use someone else's IP address and pretend to be the other respondent.

After an attacker identifies a pair of computers—A and B, for example—that are communicating with each other as a client-server pair, he attempts to establish a connection with computer B in such a way that B believes that it has a connection with A; in reality, the connection is with the attacker's computer.

The attacker accomplishes this by creating a fake message (i.e., a message from the attacker) but with A's address as the source address, requesting a connection to B. When it receives this message, B will respond with an acknowledgment, which includes sequence numbers for transmissions with A. These sequence numbers from server B are unique to the connection between the two machines.

To complete the setup of this session between A and B, B would expect A to acknowledge B's sequence number before proceeding with any further exchange of information. But, in order for the attacker to impersonate A, he has to guess the sequence numbers B will use, and he has to prevent A from replying. It turns out that, in certain circumstances, it's not too difficult to guess what the sequence numbers are.

In order to keep computer A from responding to any of B's transmissions (and thus denying that it had requested a connection in the first place), the attacker usually transmits a large number of packets to A, overflowing A's capacity to process them and preventing A from responding to B's message.

Even with automated tools, IP spoofing can be rather tedious to accomplish. Spoofing is relatively easy to protect against: Configure your routers to reject any inbound packets that claim to originate from a computer within your internal network, which prevents any external computer from taking advantage of session relationships within the internal network. If you have such relations that cross the network's borders, such as over the Internet, then guarding against IP spoofing is more difficult.

Previous | Table of Contents | Next

Session Hijacking

Spoofing is one level of attack; it makes possible another. In *session hijacking,* rather than attempting to initiate a session via spoofing, the attacker attempts to take over an existing connection between two computers.

The first step in this attack is for the attacker to take control of a network device on the LAN, either a firewall or another computer, so that he can monitor the connection. By monitoring the connection between the two computers, the attacker can determine the sequence numbers used by both parties.

After he's monitored the connection and determined the sequence numbers, the attacker can generate traffic that appears to come from one of the communicating parties, stealing the session from one of the individuals involved. As in IP spoofing, the attacker would overload one of the communicating computers with excess packets so that it drops out of the communications session.

The problems caused by session hijacking point out the need for a reliable means of identifying the other party in a session. The fact that you've identified the person with whom you're communicating once doesn't mean that you can depend on IP to ensure it will be the same person through the rest of the session. You need a scheme that authenticates the data's source throughout the transmission. Even the strongest authentication methods are not always successful in preventing hijacking attacks; the only true defense against such attacks is the widespread use of encryption.

Electronic Eavesdropping or Sniffing

Sniffing is another attack that's possible in shared-media networks like Ethernet-based IP networks. In most Ethernet LANs, packets are available to every Ethernet node on the network. The usual convention is for each node's *network interface card* (NIC) to only listen and respond to packets specifically addressed to it. It's relatively easy, however, to put many Ethernet NICs into what's called *promiscuous mode*—meaning that they can collect every packet that passes on the wire. Such a NIC cannot be detected from another location on the network, because the NIC doesn't do anything to the packets when it collects them.

A type of software colloquially called a *sniffer* (after the original network analysis tool designed to do this—Network General's Sniffer) can take advantage of this feature of Ethernet technology. Such tools can record all the network traffic going past them. As such, they are a necessary part of the toolkit of any network diagnostician working with Ethernets, allowing them to determine quickly what's going through any segment of the network. However, in the hands of someone who wants to listen in on sensitive communications, a sniffer is a powerful eavesdropping tool. For instance, an attacker can use a packet

sniffer to record all login packets on a network and then use the login information to enter systems that he would otherwise be unauthorized to access.

Sniffing also can be used to collect company data and messages as they're transmitted on a network, for later analysis. For example, the attacker might perform a traffic analysis to learn who's communicating with whom, which could be competitive intelligence on secret partnerships or merger talks, for instance.

Strong authentication using one-time passwords or tokens is one way of keeping anyone with a sniffer from reusing a password that he's illegally obtained. Encrypting data is another way of protecting your data against sniffing, although even that isn't a foolproof solution; the attacker may have the resources to store the encrypted data and try decrypting the messages off-line.

Physical inspection of your networks is a good way to reduce the risk of sniffing, because sniffers have to be physically attached to your network to intercept packets. Also, on some computers, like those running Unix, you easily can check to see whether a NIC is set to run in promiscuous mode.

The Man-in-the-Middle Attack

Although it seems obvious that using encryption technologies to conceal and authenticate the data passed in IP packets is a solution to many of the IP security threats we've just discussed, encryption is not a foolproof solution. You still need to carefully manage your encryption system to guard against other attacks, such as *man-in-the-middle attacks*.

To use encryption, you first have to exchange encryption keys. But, exchanging unprotected keys over the network could easily defeat the whole purpose of the system, because those keys could be intercepted and open your data up to yet another type of attack—the man-in-the-middle attack. A sophisticated attacker employing spoofing, hijacking, and sniffing could actually work his way into such a key exchange, in a system that left the way open. He could plant his own key early in the process so that, while you believed you were communicating with one party's key, you actually would be using a key known to the man-in-the-middle.

Types of Authentication

Authentication can be divided into two types: weak and strong authentication. Weak, or simple, authentication mechanisms are "normal" mechanisms used by most systems, for example, the use of a password when a user logs in to a system. Strong authentication mechanisms are mechanisms where an entity does not reveal any secrets during the authentication process.

The bottom line is that you need to carefully deploy and maintain your security system, and check it regularly, to ensure that it's still effective against all kinds of threats. For VPNs, two important building blocks of secure systems are *authentication* and *encryption.* Let's start out exploring different methods for authenticating users and computers and then move on to encryption and some related aspects of modern-day cryptography.

Authentication Systems

Authentication is a vital part of a VPN's security structure. Unless your system can reliably authenticate users, services, and networks, you cannot control access to your corporate resources and keep unauthorized users out of your networks.

Authentication is based on one of the following three attributes: something you have (a key to a door or a token card); something you know (a password); or something you are (voiceprints, retinal scans). It's generally accepted among security experts that a single method of authentication, such as a password, is not adequate for protecting systems. Instead, they recommend what's called *strong authentication,* or using at least two of the preceding attributes for authentication.

The variety of VPN systems currently available depend on different methods of authentication or combinations of them. As background for the following chapters, in which we discuss the details of these systems, we'll review the more common authentication methods. They'll be classified in the following way: traditional passwords, one-time passwords (S/Key), other password systems (PAP, CHAP, TACACS, and RADIUS), hardware-based (tokens, smart cards, and PC cards), and biometric IDs (fingerprint, voice print, and retinal scans).

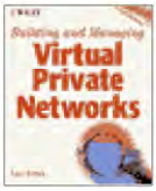
Traditional Passwords

It's generally recognized that the simplest form of authentication (i.e., user IDs and passwords) is inadequate for securing network access. Passwords can be guessed and intercepted during network transmissions.

Even when users are careful about guarding their passwords, they may not realize that different Internet services offer no protection for their passwords. For example, services such as FTP and telnet transmit user IDs and passwords as plaintext, making them easy to use when intercepted.

One-time password systems, which restrict the validity of a password to a single session, can be a good solution to some of the problems surrounding traditional password uses. We'll see shortly that some improved authentication methods choose to encrypt user IDs and passwords.

Previous | Table of Contents | Next



Building and Managing Virtual Private Networks
by Dave Kosiur
Wiley Computer Publishing, John Wiley & Sons, Inc.
ISBN: 0471295264 **Pub Date:** 09/01/98

One-Time Passwords

One way to prevent the unauthorized use of an intercepted password is to prevent it from being reusable (i.e., restrict a password's use to a single communications session). As you'd expect from the name, one-time password systems aim to do just that, by requiring a new password for each new session. These systems, of which S/Key (originally developed by Bellcore) is the best example, relieves the user of the difficulty of always choosing a new password for the next session by automatically generating a list of acceptable passwords for the user. The IETF has taken on the task of standardizing S/Key; see their specifications for the One-Time Password (OTP) System in RFC 2289.

S/Key uses a secret pass-phrase, generated by the user, to generate a sequence of one-time passwords. The user's secret pass-phrase never travels beyond his local computer and does not travel on the network; therefore, the pass-phrase is not subject to replay attacks. Also, because a different one-time password is generated for each session, an intercepted password cannot be used again, nor does it give the hacker any information about the next password to be used.

A sequence of one-time passwords is produced by applying a secure hash function multiple times to the message digest produced in the initial step. (See the section, "An Introduction to Cryptography," later in this chapter for an explanation of hash functions and message digests.) In other words, the first one-time password is produced by passing the message digest through the hash function N times, where N is specified by the user. The next one-time password is generated by passing the message digest through the hash function N–1 times, and so on, until N one-time passwords are generated.

When a user attempts to log into a network, the network server, which is the S/Key-enabled host guarding the entrance to the network, issues a challenge consisting of a number and a string of characters, which is called the *seed.*

In responding to the network server's challenge, the user enters the challenge number and seed plus his own secret pass-phrase into the S/Key generator software that runs on his computer. The generator software then combines the secret pass-phrase with the seed and iterates a hash function repeating the operation for the number of times corresponding to the challenge number. The result of the calculation is a one-time password that takes on the form of six English words.

The one-time password is sent to the network server, which also iterates the hash function and compares the result with the stored one-time password that was used for the most recent login. If they match, the user is allowed to log in. The challenge number is decremented, and the latest one-time password is kept for the next login attempt.

One-time password systems like S/Key require that the server software be modified to perform the

required calculations and that each remote computer have a copy of the client software. These systems may not be highly scalable because it's difficult to administer the password lists for a large number of users.

Other Systems

Aside from the traditional password method for authentication, which often includes sending the user ID and password in plaintext, a number of other important password-based systems have been developed for authentication, especially for remote access. Because many of the VPN systems use these methods for controlling remote access, it's worthwhile to review them briefly here. The methods are PAP, CHAP, TACACS, and RADIUS.

PASSWORD AUTHENTICATION PROTOCOL (PAP)

PAP, or the *Password Authentication Protocol,* was originally designed as a simple way for one computer to authenticate itself to another computer when *Point-to-Point Protocol* (PPP) is used as the communications protocol. PAP is a two-way handshaking protocol; that is, the host making the connection sends a user ID and password pair to the target system with which it's trying to establish a connection, and then the target system (the authenticator) acknowledges that the computer is authenticated and approved for communication.

PAP authentication can be used at the start of the PPP link as well as during a PPP session to reauthenticate the link.

When the PPP link is established, PAP authentication can be carried out over that link. The peer sends a user ID and a password in the clear to the authenticator until either the authenticator accepts the pair or the connection is terminated. PAP is not secure because authentication information is transmitted in the clear, and nothing protects against playback attacks or excessive repetition by attackers trying to guess a valid password/user ID pair.

CHALLENGE HANDSHAKE AUTHENTICATION PROTOCOL (CHAP)

CHAP was designed for the same uses as PAP, but CHAP is a more secure method for authenticating PPP links. CHAP is a three-way handshaking protocol. Like PAP, CHAP can be used at the start of a PPP link and then repeated after the link has been established.

CHAP is referred to as a three-way handshake protocol because it incorporates three steps to produce a verified link after the link is first initiated, or at any time after the link has been established and verified. Instead of a simple two-step password/approval process, such as that used by PAP, CHAP uses a one-way hashing function in a fashion similar to that used by S/Key. The actual process is as follows (see Figure 4.3):

1. The authenticator sends a challenge message to the peer.
2. The peer calculates a value using a one-way hash function and sends it back to the authenticator.
3. The authenticator can acknowledge authentication if the response matches the expected value.

The process can be repeated at any time during the PPP link to ensure that the connection has not been

taken over or subverted in any way. Unlike PAP, which is driven by the client side, the server controls CHAP reauthentication. CHAP also removes the possibility, inherent in PAP, that an attacker can try repeatedly to log in over the same connection. When the CHAP authentication fails, the server is required to drop the connection. This complicates the attacker's task of guessing the password because he cannot try new guesses in a single connection.

PAP and CHAP do have some disadvantages. Both PAP and CHAP rely on a secret password that must be stored on the remote user's computer and the local computer. If either computer comes under the control of a network attacker, then the secret password is compromised. Also, with CHAP or PAP authentication, you cannot assign different network access privileges to different remote users who use the same remote host.

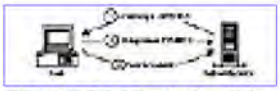


FIGURE 4.3 Challenge-response system using CHAP.

Because one set of privileges is assigned to a specific computer, everybody who uses that computer will have the same set of privileges. The next two protocols we'll discuss, TACACS and RADIUS, provide more flexibility for assigning access privileges.

Although CHAP is a stronger method than PAP for authenticating dial-up users, CHAP may not meet the scalability requirements of large organizations. Even though it doesn't transmit any secrets across a network, it requires a large number of shared secrets to be run through the hash function. Organizations with many dial-up users have to maintain very large databases to accommodate them all.

Previous | Table of Contents | Next

TERMINAL ACCESS CONTROLLER ACCESS-CONTROL SYSTEM (TACACS)

TACACS is one of the systems developed to not only offer authentication, but also add the other two As of remote access security—*authorization* and *accounting*. (Admittedly, PAP and CHAP also offer authorization or access control, but they're of limited flexibility.) Unlike the peer relations designed into PAP and CHAP, TACACS is designed to function as a client/server system, which affords it more flexibility, especially in security management. (We'll shortly see that RADIUS also is a client/server architecture.) Central to the operation of TACACS, and RADIUS, is an authentication server (see Figure 4.4).

Typically, a TACACS authentication server handles requests from authentication client software that's installed at a gateway or network entry point. The authentication server maintains a database of user IDs, passwords, PINs, and secret keys, which it uses to grant or deny network access requests. All authentication, authorization, and accounting data is directed to the centralized server when a user tries to log in.

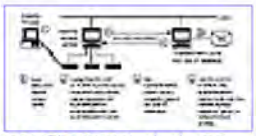


FIGURE 4.4 How authentication servers authorize remote access.

TACACS transmits all data in the clear between the user and the server, but a recent update from Cisco, TACACS+, adds a message-digest function to eliminate the plaintext transmission of passwords. TACACS+ also supports multiprotocol logins, meaning that a single user ID and password pair can authenticate a user for multiple devices and networks—for example, an IP network login and an IPX network login. Finally, TACACS+ also can handle PAP and CHAP authentication.

TACACS is currently best known as Cisco System's server-based security software protocol. All Cisco router and access-server product families use this protocol. Although TACACS has been described in an IETF RFC, and is freely available for other vendors to implement, most vendors view TACACS as proprietary and instead concentrate on RADIUS.

One advantage to TACACS is that it can act as a proxy server to other authentication systems, such as a Windows NT security domain, NDS, Unix-based NIS maps, or other security systems (such as the token-based systems we'll mention shortly). The proxy capabilities also make it easier for a corporate client to share VPN security data with an ISP, which is necessary when a VPN is outsourced; the ISP runs a proxy server to control dial-in access based on access rights managed by the corporate customer on its own secure server. But, transmitting authentication packets between the parent server and the

proxy server across a public network poses a security risk. RADIUS and TACACS encryption is based on static keys; the user names, passwords, and authentication server info are conveniently contained in a single packet, making them easier to use if intercepted.

REMOTE AUTHENTICATION DIAL-IN USER SERVICE

The *Remote Authentication Dial-In User Service* (RADIUS) protocol also uses a client/server model to securely authenticate and administer remote network connection users and sessions. RADIUS is largely a way to make access control more manageable, and it can support other types of user authentication, including PAP and CHAP.

The RADIUS client/server model uses a *network access server* (NAS) to manage user connections. Although the NAS functions as a server for providing network access, it also functions as a client for RADIUS (see Figure 4.4). The NAS is responsible for accepting user connection requests, getting user ID and password information, and passing the information securely to the RADIUS server. The RADIUS server returns authentication status—approved or denied—as well as any configuration data required for the NAS to provide services to the end user.

RADIUS clients and servers communicate securely, using shared secrets for authentication and encryption for transmitting user passwords.

RADIUS creates a single, centrally located database of users and available services, a feature particularly important for networks that include large modem banks and more than one remote communications server. With RADIUS, the user information is kept in one location, the RADIUS server, which manages the authentication of the user and access to services from one location. Because any device that supports RADIUS can be a RADIUS client, a remote user will gain access to the same services from any communications server communicating with the RADIUS server.

Hardware-Based Systems

Earlier, when we wrote about the different methods for authentication, we mentioned that one class of methods focuses on using something that you have in your possession. This is where hardware devices come into play, such as smart cards, PC cards, and token devices.

SMART CARDS AND PC CARDS

Smart cards are devices about the size of a credit card but include an embedded microprocessor and memory. A smart-card terminal or similar reader for smart cards is required to communicate with a smart card so that information can be exchanged as needed. Many of these readers are now available for use with a PC floppy drive or are integrated into keyboards, making their use with PCs simpler than before.

Smart cards can store a user's private key along with any installed applications, which simplifies the authentication process, especially for mobile users. Some smart cards now include their own cryptographic coprocessors, making encryption of data easier and faster than with older smart cards. And, many software developers are now taking advantage of standardized APIs, like the CryptoAPI for use with Windows, to tie together smart cards and PCs.

The simplest systems for using digital certificates require the user to enter a PIN to complete the

authentication process. In some cases, a PIN is stored on the smart card, and use of the PIN to authenticate the user is checked automatically by the smart card before any other communication with the rest of the system takes place. When a PIN isn't stored on the card, this method may not be secure enough (PINs can be guessed), so higher end systems combine the information stored in the smart card with biometric information. Using these systems, the card reader includes a biometric device, such as a fingerprint scanner. The scanned data then is compared with the data stored on the smart card to authenticate the card holder. This process soon may happen entirely on a smart card, as Verdicom and Lucent Technologies recently announced the development of a fingerprint scanner on a chip that can be installed on a smart card.

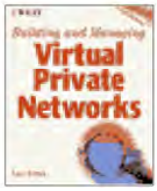
Although smart cards are seeing increasing use in security systems, it's also possible to use other types of electronic cards that can be inserted into a PC. One example is the PC card. PC cards, which used to be called PCMCIA cards, are those small circuit boards that can be inserted into special slots on desktop computers, and particularly laptops, to provide added functionality. These cards can offer some of the same functionality as smart cards but are restricted to use with PCs containing PCMCIA slots, making them less portable if a variety of access devices are to be used. However, PCMCIA cards do have the advantage of more memory, enabling them to store larger files for authentication purposes.

TOKEN DEVICES

Token-based systems usually are based on separate hardware (i.e., not built into a PC) that displays changing passcodes that a user then has to type into his computer for authentication.

Here's a quick rundown on how token-based authentication works. A processor inside the token card stores a series of secret encryption keys used to generate one-time passcodes. The passcodes are sent to a secure server on the network, which checks their validity and grants the user access. After the codes are programmed in, neither users nor administrators have access to them.

Previous | Table of Contents | Next



Building and Managing Virtual Private Networks
by Dave Kosiur
Wiley Computer Publishing, John Wiley & Sons, Inc.
ISBN: 0471295264 **Pub Date:** 09/01/98

Before users are permitted to authenticate themselves, token devices request a PIN. They then use one of three different mechanisms to verify that users are who they say they are. The most widely implemented of these mechanisms is challenge-response (see Figure 4.3), in which the secure server issues a random number (called the *challenge*) when the user attempts to log in. The challenge appears on the screen of the user, who then types the numbers into the token card. The card encrypts the challenge with its secret key and displays the response on its LCD screen, and the user then types that response into the PC. Meanwhile, the server encrypts the challenge with the same key, and if the two results match, the user is allowed in.

Another scheme makes use of time synchronization. Here, the token displays a number encrypted with the secret key, which changes every 60 seconds. Users are prompted for the number when they try to log into the server. Because the clocks on the server and the token are synchronized, the server can authenticate the user by decrypting the token number and comparing results. (Users caught typing during the middle of a passcode change usually have to start again with the new code.)

The third scheme is event synchronization, a variation on time synchronization. Here, a counter records the number of login attempts made by a user. After every attempt, the counter is updated, and a different passcode is generated for the next login.

Problems with token-based systems stem from their use of extra hardware and the involvement of a human being to enter the authentication codes. This latter point not only can prove to be tedious for the user, but also makes authentication of unattended batch applications impossible.

Biometric Systems

Biometrics depends on using a unique personal trait to identify the user. You've probably seen a James Bond movie or other spy story in which voice prints, retinal scans, and hand images were used to identify (or misidentify) a main character of the film. Biometric technologies measure human characteristics such as fingerprints, voice recordings, iris and retinal scans, heat patterns, facial images, and even keystroke patterns. But, biometric systems have yet to see routine use in many environments because they've been expensive and usually are all-in-one security systems, making them difficult to interface with other security systems. That's likely to change, however, as newer, faster, and less expensive technologies come into play.

One approach that's likely to see widespread deployment is fingerprint scanning. Fingerprint scanners have dropped in price considerably and are being incorporated into PC keyboards in 1998. Also, as mentioned, a scanner-on-a-chip has been developed that enables fingerprint scanning to be directly incorporated into a smart card.

Some of the newer face analysis systems can operate on a PC with a low-cost, low-resolution camera such as is often used for videoconferencing. A central database stores images of authorized users and compares the image transmitted by the camera to the stored images to grant access.

Although the use of biometric systems appears to be on the rise, the lack of a standardized set of Application Programming Interfaces (APIs) for most of the biometric methods makes it difficult to readily incorporate biometrics into existing security systems. At least four different APIs have been proposed for developing security applications: the Biometric API (BAPI), backed by Japanese hardware manufacturers; the Human Authentication API (HA-API), developed by National Registry Inc. for the Department of Defense; the Speaker Verification API (SVAPI), which has been proposed for voice recognition systems; and an API proposed by IBM. In an attempt to promote a common API for biometrics, Compaq Computer, IBM, Identicator Technology, Microsoft, Miros, and Novell formed the BioAPI Consortium in April, 1998.

An Introduction to Cryptography

Modern-day cryptographic algorithms coupled with today's powerful microprocessors now make possible the everyday use of powerful authentication and encryption methods. Cryptography covers a number of algorithms for encrypting and decrypting information, classified according the way secrets, or keys, are shared between correspondents, how the secrets are used to encrypt and decrypt information, and what form the algorithms take. For a complete review of cryptography, see *Applied Cryptography* by Bruce Schneier (2d edition, John Wiley & Sons, Inc., 1996); this chapter only covers a few cryptographic algorithms that are particularly pertinent to network security and VPNs.

What Is Encryption?

Encrypting or encoding information to prevent its being read by unauthorized parties has been the main use of cryptography since its early beginnings—Julius Caesar, for instance, used an alphabetical cipher when communicating with his field commanders.

For encryption to work properly, both the sender and receiver have to know what set of rules, called the *cipher,* was used to transform the original information into its coded form, often called *cipher text*. A simple cipher might be to add an arbitrary number of characters, say 13, to all characters in a message. As long as the receiving party knows what the sender did to the message, the receiving party can reverse the process (for example, subtract 13 characters from the message received) to extract the original text.

Encryption is based on two components: an algorithm and a key. A cryptographic algorithm is a mathematical function that combines plaintext or other intelligible information with a string of digits called a *key* to produce unintelligible cipher text. The key and the algorithm used are both crucial to the encryption.

Although some special encryption algorithms that don't use a key do exist, algorithms using keys are particularly important. (See the discussion of hash functions in the section, "What Is Public-Key Cryptography?," later in this chapter.) Basing encryption on a key-based system offers two important advantages. First, encryption algorithms are difficult to devise; you wouldn't want to come up with a new algorithm each time you want to communicate privately with a new correspondent. By using a key, you can use the same algorithm to communicate with many people; all you have to do is use a different key

for each correspondent. Second, if someone does crack your encrypted messages, all you have to do is switch to a new key to start encrypting messages all over again; you don't have to switch to a new algorithm (unless the algorithm and not the key proved to be insecure—that can happen, but it's unlikely).

The number of possible keys each algorithm can support depends on the number of bits in the key. For example, an 8-bit key length allows for only 256 (2^8) possible numeric combinations, or keys. The greater the number of possible keys, the more difficult it is to crack an encrypted message. The level of difficulty is therefore dependent on the key length. It would not take a computer very long to sequentially guess each of the 256 possible keys (less than a millisecond) and decrypt the message to see whether it makes sense. But, if a 100-bit key were used, which equates to searching 2^{100} keys, and the computer could guess 1 million keys every second, it could actually take many centuries to discover the right key.

Previous | Table of Contents | Next

The security of an encryption algorithm correlates with the length of its key. Why? Because knowing that a key is *n* bits long only gives you an idea of how much time you'd have to spend to break the code. If security were dependent on such things as the secrecy of the algorithm, or the inaccessibility of the cipher text or plaintext, unauthorized persons could derive that information from publications or the pattern analysis of messages, or they could collect the information in other ways (traffic monitoring, for example). When the information is in hand, the unauthorized person(s) can use it to decrypt your communications.

The oldest form of key-based cryptography is called *secret-key* or *symmetric* encryption. In this scheme, both the sender and recipient possess the same key, which means that both parties can encrypt and decrypt data with the key (see Figure 4.5). But, symmetric encryption has some drawbacks: for example, both parties must agree upon a shared secret key. If you have *n* correspondents, then you have to keep track of *n* secret keys—one for each of your correspondents. If you use the same key for more than one correspondent, then they will be able to read each other's mail.

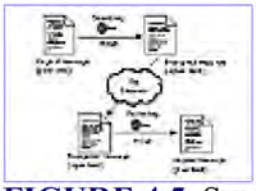


FIGURE 4.5 Symmetric encryption uses a single secret key to encrypt and decrypt messages.

Symmetric encryption schemes also have a problem with authenticity, because the identity of a message's originator or recipient cannot be proved. Because both Ann and Tim possess the same key, both of them can create and encrypt a message and claim that the other person sent it. This built-in ambiguity about who authored a message makes nonrepudiation impossible with secret keys. Proving that someone actually did send a message when he claims he didn't is called *nonrepudiation.* The way to solve the repudiation isssue is by using what is called public-key cryptography, which makes use of asymmetric encryption algorithms.

What Is Public-Key Cryptography?

Public-key cryptography is based on the concept of a key pair. One part of the key pair, the *private key,* is known only by the designated owner; the other part, the *public key,* can be published widely but is still associated with the owner. Key pairs have a unique feature: Data encrypted with one key can be decrypted with the other key in the pair (see Figure 4.6). You'll see some of the power of this in the next few pages.

These keys can be used in two different ways: to provide message confidentiality and to prove the authenticity of a message's originator. In the first case, the sender uses the recipient's public key to encrypt a message so that it will remain confidential until decoded by the recipient with the private key. In the second instance, the sender encrypts a message using the private key, a key to which only the sender has access.

FIGURE 4.6 Using a key pair to encrypt and decrypt a message.

For example, in order to create a confidential message, Tim first would acquire Ann's public key. Then he uses her public key to encrypt the message and sends it to her. Because the message was encrypted with Ann's public key, only someone with Ann's private key (and presumably only Ann has that) can decrypt the message.

Although encrypting a message with part of a public key pair isn't very different from using secret-key encryption, public-key systems offer some advantages. For instance, the public key of your key pair can be readily distributed (on a server, for example) without fear that this compromises your use of your private key. You don't have to send a copy of your public key to all your respondents; they can get it from a key server maintained by your company or maybe a service provider.

Another advantage of public-key cryptography is that it enables you to authenticate a message's originator. The basic idea is this: Because you are the only person who can encrypt something with your private key, anyone using your public key to decrypt the message can be sure that the message came from you. Thus, your use of your private key on an electronic document is similar to your signing a paper document. The recipient then will be certain that the message came from you but cannot be sure that nobody else has read it as well.

Using public-key cryptographic algorithms to encrypt messages is computationally slow, so cryptographers have come up with a way to quickly generate a short, unique representation of your message, called a *message digest,* which can be encrypted and then used as your digital signature.

Some popular, fast cryptographic algorithms for generating message digests are known as one-way hash functions. A one-way hash function doesn't use a key; it's simply a formula to convert a message of any length into a single string of digits called a message digest. When using a 16-byte hash function, text processed with that hash function would produce 16 bytes of output. A message might result in the string, for example "CBBV235ndsAG3D67". The important thing to remember is that each message produces a random message digest.

Message digests on their own can prove useful as an indicator that data hasn't been altered, but digital signatures are even more reliable. If you encrypt the message digest with your private key, you've got a digital signature.

As an example, let's have the sender, Tim, calculate a message digest for his message, encrypt the digest with his private key, and send that digital signature along with the plain-text message to Ann (see Figure

4.7).

After Ann uses Tim's public key to decrypt the digital signature, she has a copy of the message digest that Tim calculated. Because she was able to decrypt the digital signature with Tim's public key, she knows that Tim created it, authenticating the originator. Ann then uses the same hash function, which was agreed-upon beforehand, to calculate her own message digest of Tim's plain-text message. If her calculated value and the one Tim sent her are the same, then she can be assured that the digital signature is authentic, which means that Tim sent the message and the message itself has not been tampered with.

The one problem with this approach is that a copy of the plaintext is sent as part of the message and, therefore, privacy is not maintained (i.e., someone could still read the data even if they couldn't alter it). If you want to maintain the data's privacy, you should encrypt the message. But, to reduce the computational overhead, use a symmetric algorithm with a secret key. This procedure further complicates matters, but it might be worth the added work.

Two Important Public-Key Methods

Although a wide variety of cryptographic algorithms exist for public keys, two public-key methods—Diffie-Hellman and RSA—account for the majority of public-key usage these days.

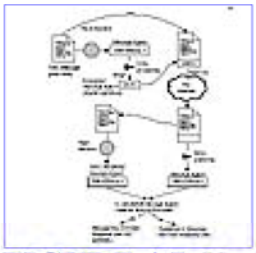


FIGURE 4.7 Verifying a digital signature.

THE DIFFIE-HELLMAN TECHNIQUE

The Diffie-Hellman technique was the first practical public-key cryptographic algorithm; in practice, Diffie-Hellman is very useful for key management. We'll see in the next chapter that the key exchange proposals for IPSec are each based on Diffie-Hellman.

On to the mechanics...Two correspondents can use Diffie-Hellman to produce a shared secret value that then can be used as a common key for a secret key encryption algorithm (see Figure 4.8). Let's have Tim and Ann each generate a random number on their computers; these two random numbers become their private keys. In order to communicate, they first exchange some public data that is considered their public key. Ann then applies her private key to Tim's public key to compute the shared secret value. Tim does likewise, applying his private key to Ann's public key, computing the same value.

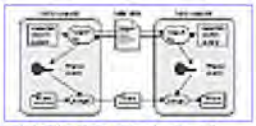


FIGURE 4.8 Producing a Diffie-Hellman shared secret.

Should someone intercept the public values, they cannot easily compute the random secret values from them. The crucial point of the Diffie-Hellman algorithm is that Ann and Tim will both end up with the same numerical result, and nobody else can easily compute the same result from the publicly available information.

Basically, Diffie-Hellman works because you can apply exponentiation in different orders and still get the same result. In Diffie-Hellman, both Ann and Tim agree on a particular base number. That base number raised to the power of an individual's private key, a large random number, becomes the public key, say B^A (i.e., $\text{Base}^{\text{Ann's\_key}}$) for Ann. Tim's public key would be B^T. Now, when Tim receives Ann's public key, that is, B^A, he can raise it to the power of his private key to get the shared secret (i.e., $(B^A)^T$). When Ann receives Tim's public key, B^T, she can raise it to the power of her private key to get $(B^T)^A$, which is identical to the other result that Tim calculated.

The Diffie-Hellman technique can be particularly useful for creating temporary session keys, which are used only by the corresponding parties during an exchange of information and are deleted afterwards. Using a new key for each session reduces the risk of compromising your security.

Perfect Forward Secrecy

One of the reasons for continued interest in Diffie-Hellman is because it can be used to achieve perfect forward secrecy. The more often you use the same key to encrypt data, the greater the risk of having the key compromised. Longer keys help the situation somewhat, so picking some reasonable length that doesn't impose severe performance slowdowns and changing the keys frequently can reduce the risk. But, because each new key cannot be related to any previous keys (or an attacker will have more useful information to help crack the key), you need a method of generating a new key that's independent of the value of the current key. Diffie-Hellman makes that possible; cryptographers call the concept *perfect forward secrecy*.

RSA PUBLIC-KEY CRYPTOGRAPHY

The RSA public-key technique derives its name from its three developers: Ron Rivest, Adir Shamir, and Leonard Adelman. The security of this approach is based on the fact that it can be relatively easy to multiply large prime numbers together, but it is almost impossible to factor the resulting product. This

technique produces public keys that are tied to specific private keys. This gives RSA the advantage of enabling the holder of a private key to encrypt data with it so that anyone with a copy of the public key can then decrypt it, much as we explained in the beginning of the section on public-key cryptography.

RSA keys consist of three special numeric values that are used in pairs to encrypt or decrypt data. The RSA public key consists of a public-key value (normally either 317 or 65,537) and a modulus. The modulus is the product of two large prime numbers, chosen at random, that are mathematically related to the chosen public key. The private key is calculated from the two prime numbers that were generated for the modulus and the public-key value.

In practice, the private key cannot be derived because there is no practical way to compute the values of the two selected prime numbers by factoring the modulus.

Selecting Encryption Methods

No one encryption system is ideal for all situations. Table 4.1 illustrates some of the advantages and disadvantages of each type of encryption.

TABLE 4.1 Advantages and Disadvantages of Cryptographic Systems

Encryption Type	*Advantages*	*Disadvantages*
Symmetric Key.	Fast.	Both keys are the same.
	Can be easily implemented in hardware.	Difficult to distribute keys.
		Does not support digital signatures.
Public Key.	Uses two different keys.	Slow and computationally intensive.
	Relatively easy to distribute keys.	
	Provides integrity and nonrepudiation through digital signatures.	

When selecting an appropriate algorithm to use, the general rule of thumb is this: First, determine how sensitive your data is and for how long it will be sensitive and have to be protected. When you've figured that out, select an encryption algorithm and key length that will take longer to break than the length of time for which your data will be sensitive.

One of the best discussions of key lengths and the efforts required to break a key is found in Chapter 7 of *Applied Cryptography* by Bruce Schneier (2nd edition, John Wiley & Sons, Inc., 1996). Table 4.2 is a condensation of his table estimating the cost of building a computer in 1995 to crack symmetric keys and the time required to crack certain length keys.

Remember that this is not a static situation. Computing power is always going up, and costs are falling, so it'll get easier and cheaper to break larger keys in the future. These estimates are for brute-force attacks—that is, guessing every possible key. There are other methods for cracking keys, depending on

the ciphers used (that's what keeps cryptanalysts employed), but estimates for brute-force attacks are commonly cited as a measure of the strength of an encryption method. For further details, see Bruce Schneier's Web site at www.counterpane.com.

Previous | Table of Contents | Next

TABLE 4.2 Comparison of Time and Money Needed to Break Different Length Keys

	Length of key in bits				
Cost	***40***	***56***	***64***	***80***	***128***
$100 K	2 secs.	35 hrs.	1 yr.	70,000 yrs.	1019 yrs.
$1 M	.2 secs.	3.5 hrs.	37 days	7000 yrs.	1018 yrs.
$100 M	2 millisecs.	2 mins.	9 hrs.	70 yrs.	1016 yrs.
$1 G	.2 millisecs.	13 secs.	1 hr.	7 yrs.	1015 yrs.
$100 G	2 microsecs.	.1 sec.	32 secs.	24 days	1013 yrs.

Common Key Algorithms

DES (Data Encryption Standard). A block cipher created by IBM and endorsed by the U.S. government in 1977. Uses a 56-bit key and operates one block of 64 bits. Relatively fast and used to encrypt large amounts of data at one time.

Triple DES. Based on DES. Encrypts a block of data three times with three different keys. Being proposed as an alternative to DES, because it's been said that the potential of easily and quickly cracking DES is increasing every day.

RC2 and RC4. Designed by Ron Rivest (the *R* in RSA Data Security Inc.). Variable key-size ciphers for very fast bulk encryption. A bit faster than DES, the two algorithms can be made more secure by selecting a longer key size. RC2 is a block cipher and can be used in place of DES. RC4 is a stream cipher and is as much as 10 times faster than DES.

IDEA (International Data Encryption Algorithm). Created in 1991, it was designed to be efficient to compute in software. Offers very strong encryption using a 128-bit key.

RSA. Named after Rivest, Shamir, and Adelman, its designers. Public-key algorithm supports a variable key length as well as a variable blocksize of the text to be encrypted. The plaintext block must be smaller than the key length. Common key length is 512 bits.

Diffie-Hellman. The oldest public-key cryptosystem still in use. Does not support either encryption or digital signatures. System is designed to allow two individuals to agree on a shared key, even though they only exchange messages in public.

DSA. Digital Signature Algorithm, developed by NIST and based on what's called the El Gamal algorithm. The signature scheme uses the same sort of keys as Diffie-Hellman and can create signatures faster than RSA. Being pushed by NIST as DSS, the Digital Signature Standard, although its acceptance is far from ensured.

Blowfish. A 64-bit block cipher with a variable-length key designed by Bruce Schneier for implementation on large microprocessors. It's optimized for applications in which the key does not change often.

Skipjack. The NSA-developed encryption algorithm designed for the Clipper and Capstone chips. The algorithm is an iterative 64-bit block cipher with an 80-bit key.

Secret- and public-key ciphers use different key lengths, so the preceding table cannot be used for setting all of your security requirements. Table 4.3 compares the two systems for similar resistance to brute-force attacks.

TABLE 4.3 Secret-Key and Public-Key Lengths for Equivalent Levels of Security

Secret-Key Length	*Public-Key Length*
56 bits	384 bits
64 bits	512 bits
80 bits	768 bits
112 bits	1,792 bits
128 bits	2,304 bits

When it comes to selecting software or hardware for your purposes, recall that more than one encryption system might be used in the product—that's a common practice because of the different computational requirements for secret-key and public-key algorithms. For example, basic implementations of IPSec use a keyed MD5 hash function for authentication of packets and DES for data encryption; other encryption techniques, such as RC4, can be negotiated between IPSec partners.

Public-Key Infrastructures

Although we've spent a lot of time describing how authentication and encryption can be used and what roles secret and public keys play, we've said very little about how these keys are generated and distributed. The security services that make this possible fall under the umbrella term Public Key Infrastructure (PKI). A PKI enables organizations to define the security domains in which they issue keys and the associated certificates, which are electronic objects used to issue and validate public keys. A PKI makes it possible not only to use keys and certificates, but also to manage keys, certificates, and security policies. Without such a system, use of public keys would be chaotic, inefficient, unmanageable, and most likely not secure.

We are not going to go into all the details of PKIs and the management of keys and certificates in this chapter but will leave the details for a later chapter in this book, Chapter 13, "Security Management." For

the moment, we'll discuss the basic concepts of public-key certificates and key generation so that you can understand how VPN systems use keys.

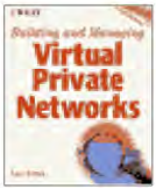
PUBLIC-KEY CERTIFICATES

Public-key certificates (see Figure 4.9) are specially formatted data blocks that tell us the value of a public key, the name of the key's owner, and a digital signature of the issuing organization, called a *certificate authority* (CA). These certificates are used to identify the owner of a particular public key.

FIGURE 4.9 Contents of a public-key certificate.

As long as you have a copy of the authority's public key, you can use it to check the certificates that it signed (see Figure 4.10). (We'll soon get to the procedures for dealing with validation of a certificate authority.) Any cryptographic software that you use must have a copy of the CA's public key in order to check a certificate's digital signature.

Previous | Table of Contents | Next



Building and Managing Virtual Private Networks
by Dave Kosiur
Wiley Computer Publishing, John Wiley & Sons, Inc.
ISBN: 0471295264 **Pub Date:** 09/01/98

The primary standard for certificates is the X.509 standard designed by the *International Telegraph Union* (ITU). This standard not only specifies the format of the certificate but also the conditions under which certificates are created and used.

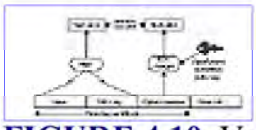


FIGURE 4.10 Validating a public-key certificate.

GENERATING PUBLIC KEYS

In order to use public-key cryptography, you need to generate a public key and a private key. After you've generated both keys, it's your responsibility to keep your private key secure and let no one else see it. Then you have to decide how to distribute your public key to your correspondents.

There are two approaches to generating public-key pairs: Some systems generate them on the host belonging to the key's holder, and others generate the keys as part of generating certificates.

First, you can generate them on the computer belonging to the key's holder, as illustrated in Figure 4.11. The user generates a public-key pair, retains the private key, and delivers the public key to the certificate authority to produce a certificate.

The second method is to have the certificate authority generate the public-key pair, produce the signed certificate, and then deliver both the key pair and the certificate to the user. Table 4.4 lists the trade-offs between these alternatives.

CERTIFICATE AND KEY DISTRIBUTION

Even though public keys are easier to distribute than secret keys, a trusted means of delivering public keys is necessary. Otherwise, it would still be possible to use a man-in-the-middle attack to trick a pair of public-key users into sharing a private communication. Aside from trusted manual distribution, the common method for delivering public keys is via digital certificates, or public-key certificates. (We'll call them certificates for short.)

FIGURE 4.11 Generating a public key.

TABLE 4.4 Advantages and Disadvantages of Key Generation Schemes

Owner-Generated Keys	*Authority-Generated Keys*
– Users must deliver key to CA.	+ Fewer steps for users to perform.
+ Private key does not need	+ Private key can be backed up.to be copied.
+ Personal signature keys do not get backed up.	+ Key generation can be shared among users.

Certificates provide a safe method of distributing public keys via electronic media. After certificates are created, the next problem is to deliver the certificates to the hosts that need them. The techniques most often used in practice are transparent distribution and interactive distribution.

Transparent distribution involves either directory servers or key exchange protocols. The directory protocols for delivering public-key certificates evolved from the X.500 directory concept originally developed to support X.400 e-mail. Although large-scale master directories for certificates may be based on X.500, there's been a significant move to use another protocol, LDAP or Lightweight Directory Access Protocol, to utilize much of the structure of X.500, but over TCP/IP. Many certificate servers now offered for use at corporate sites are based on LDAP. We'll say more about key exchange protocols in Chapter 5, "Using IPSec to Build a VPN."

Interactive distribution usually consists of either e-mail requests, access to Web sites, or requests using the finger protocol. Many e-mail systems with support for cryptography provide a way to include a certifiate with the messages they send; in some cases, a certificate server can be configured to accept e-mail requests for certificates.

CERTIFICATE AUTHORITIES

But from where does a CA get its authority? What makes it a trusted party in the scheme of things? Although there are two different types of certificate distribution systems—a hierarchical setup and a web of trust—we are going to concentrate on the hierarchical system because a web of trust isn't very scalable.

In a hierarchical system, a *root public key* exists at the top of the hierarchy, and it's used to sign for all top-level authorities, this root key might belong to a government agency, such as the DoD or the U.S. Postal Service, for example. CAs at the next lower level in the hierarchy have their certificates signed by the top-level CAs and sign for CAs below them in the hierarchy, and so on, down to the lowest level of the system.

The Lightweight Directory Access Protocol (LDAP)

Although the X.509 standard is designed for use with a globally distributed directory model, the X.500 directory standard was created for use with other ISO standards, and it's difficult to implement all of the client features on PCs using TCP/IP. The Lightweight Directory Access Protocol is a *lightweight* version of the X.500 client access *Directory Access Protocol* (DAP), which specifies how a client accesses a directory server.

LDAP can be mapped onto either proprietary services or X.500. LDAP has become a popular protocol for linking directories, and recent industry efforts have been adding many new features to LDAP, turning it into more of a directory protocol in its own right and making it less lightweight as time passes.

LDAP's extensible nature makes it appealing to use for key management, because an LDAP directory storing keys and certificates can be used both for authentication and for granting access rights based on the authentication.

To validate a user's certificate fully, you have to validate all the CAs in a hierarchy between your local CA and the issuing CA. That could include traveling up one branch of a CA hierarchy to the root and down another (see Figure 4.12).

Previous | Table of Contents | Next

In real life, CA hierarchies are not very deep—that is, they do not have many levels and sub-levels—so the time required to validate a key is short and does not seriously impact network usage. In fact, for a VPN, a corporation can serve as the CA without bothering to link to any national or international hierarchy. But, if you extend your VPN to include business partners, creating an extranet, you'll probably have to depend on some CA hierarchy for validating certificates. If the number of outside users of your VPN/extranet is small, they might agree to use your internal CA.

Using a CA hierarchy might not be a problem at the present time because the number of CAs is relatively small and hierarchies are shallow. But, as more and more uses for certificates are created and more certificates are issued, the number of CAs are bound to increase, and hierarchies will get more complicated.

Certificate authorities can offer ways to short-circuit the validation hierarchy by cross-certifying each other. If two CAs each agree to certify each other, a request for validating a certificate issued by one CA can be directly passed to the other CA without involving the rest of the CA hierarchy.

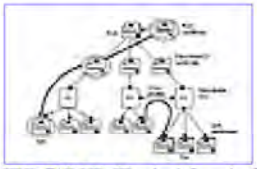


FIGURE 4.12 A hierarchy of certificate authorities.

A better, trusted way of distributing public keys is to use a certificate authority. A certificate authority will accept your public key, along with some proof of your identity (it varies with the class of certificate), and serve as the repository of a digital certificate that others can request to verify your public key. The digital certificate acts like an electronic version of a driver's license. As an accepted method for distributing your public key, it provides you with a way for correspondents to verify that you are who you say you are.

Certificate authorities, such as VeriSign, CyberTrust, and Nortel, issue digital certificates. As shown in Figure 4.9, a certificate includes the holder's name, the name of the certificate authority, a public key for cryptographic use, and a time limit for the use of the certificate (most frequently, six months to a year long).

A digital certificate can be issued in one of four classes, indicating to what degree the holder has been verified. Class 1 is the easiest to obtain because it involves the fewest checks on the user's background; only the name and e-mail address are verified. For a Class 2 certificate, the issuing authority checks a driver's license, social security number, and date of birth. Users applying for a Class 3 certificate can

expect the issuing authority to perform a credit check using a service such as Equifax in addition to the information required for a Class 2 certificate. A Class 4 certificate includes information about the individual's position within an organization, but the verification requirements for these certificates have not yet been finalized by certificate issuers.

Certificate authorities also have the responsibility of maintaining and making available a *Certificate Revocation List* (CRL), which lets users know which certificates are no longer valid. The CRL doesn't include expired certificates, because each certificate has an expiration date built-in. However, certificates may be revoked because they were lost, stolen, or because an employee left the company, for example.

In addition to commercial certificate authorities, such as VeriSign, CyberTrust, and Nortel, and government authorities, such as the U.S. Postal Service, corporations also can become a certificate authority by purchasing a certificate server from a vendor who has been certified by a certificate authority. Such arrangements are useful when a company needs to issue digital certificates to a number of employees for doing business, either within the company or with other companies. As more systems begin to use digital certificates to control computer access, corporate-maintained certificate servers will become more important. In the meantime, the U.S. government is trying to set up a Public Key Infrastructure for certificate authorities.

If a company creates its own internal CA, it has to be prepared to create key pairs, issue certificates, and manage these keys and certificates. Such a setup includes the following services:

- Public-key certificates
- A certificate repository
- Certificate revocation
- Key backup and recovery
- Support for nonrepudiation of digital signatures
- Automatic update of key pairs and certificates
- Management of key histories
- Support for cross-certification
- Client-side software

Such an arrangement isn't overwhelming, although it is does require additional resources, and some organizations have chosen to outsource the PKI management. We'll cover more of the details of PKI management in Chapter 13.

Summary

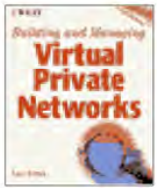
Despite the variety of threats to networked data transmissions and access to networked devices, we've seen that the combination of authentication and encryption techniques can go a long way toward thwarting network attacks.

Using CHAP and/or RADIUS for authenticating remote access users is employed commonly for PPP links and therefore has significant bearing on some of the more important systems for dial-in VPNs, particularly PPTP and L2TP, which we'll cover in Chapters 6 and 7. Other authentication methods, particularly those using hardware tokens and/or biometrics, can be deployed with existing systems, such

as RADIUS, to further improve the strength of authentication.

Encryption is fundamental to maintaining the privacy and integrity of data as it is transmitted on a network. Although secret-key encryption is easier to use and generally faster, the management of secret keys can be problematic as the number of corresponding parties grows. Public-key systems improve on the key management problem and offer additional advantages, particularly the capability to create digital signatures. However, when it becomes necessary to verify a public key or digital signature, outside organizations (certificate authorities) are required to provide validation information. Companies can serve as their own certificate authorities for VPNs because all users of the VPNs will be company employees. But, when creating an extranet, or communicating with outside correspondents that require verification, other certificate authorities likely will have to be involved.

Previous | Table of Contents | Next



Building and Managing Virtual Private Networks
by Dave Kosiur
Wiley Computer Publishing, John Wiley & Sons, Inc.
ISBN: 0471295264 **Pub Date:** 09/01/98

CHAPTER 5
Using IPSec to Build a VPN

Three major protocol suites have been proposed for building VPNs: IPSec, PPTP, and L2TP. This chapter, and the following two, concentrate on the details of these protocols and how they affect VPN design. With this coverage, you'll learn what the relative advantages and disadvantages of each technology are, which should help you pick the optimal solution for your corporation's VPN.

Each of the protocols covered in this book has some strengths and weaknesses when it comes to deploying it for VPNs. In cases like IPSec, it's more a question of being able to deploy all the features of IPSec and ensuring that all eventualities, such as key exchange, can be handled properly in the real world than it is a question of any deficiency in the protocol specifications. Plus, development continues on IPSec and related protocols as real-world examples point out what other features may need standardizing.

IPSec is the best starting point to discuss VPNs for three reasons:

1. Despite some specifications that leave implementation details up to the vendors and, therefore, leave the door open for possible interoperability problems, it offers the most complete framework for VPNs.
2. The other protocols are leaning towards using parts of IPSec for their security services.
3. Because IPSec covers both LAN-to-LAN and client-to-LAN VPNs, other protocols can be described by comparing their features to IPSec.

This chapter starts out with an overview of IPSec's architecture and moves on to the details of how the protocol works. We've also included an extensive section on key management, since it's crucial to the operation of IPSec. Then, we move on to an overview of the types of products you can use to build a VPN using IPSec.

What Is IPSec?

As mentioned before, the original TCP/IP protocols did not include any inherent security features. In the early stages of the Internet, when many of the users were academic and research institutions, the need for securing data was much less than it is today with a wide variety of commercial uses taking place on the Internet. To address the issue of providing packet-level security in IP, the IETF has been working on the IPSec protocols within their IP Security Working Group. The first protocols comprising IPSec, for authenticating and encrypting IP datagrams, were published by the IETF as RFCs 1825 to 1829 in 1995.

These protocols set out the basics of the IPSec architecture, which includes two different headers designed for use in IP packets. The IP packet is the fundamental unit of communications in IP networks, including information on the source and destination as well as the type of data being carried in the packet (see Figure 5.1). IPSec defines two headers for IP packets to handle authentication and encryption: One, the IP Authentication Header (AH), is for authentication; the other, the Encapsulating Security Payload (ESP), is for encryption purposes.

Much of the development of IPSec took place during the development of the next generation of IP protocols, now called IPv6, and was intended for inclusion in IPv6. Because of the slow adoption of IPv6 and the current need for securing IP packets, IPSec has been modified to be compatible with the IPv4 protocols as well. Support for the IPSec headers is optional for IPv4 but mandatory for IPv6. Because IPSec is compatible with IPv4, current networking applications wanting to use IPSec can do so by using special TCP/IP stacks that have been written to include the IPSec protocols. As more networks transition to IPv6 and as more IPv6 stacks become available and are deployed, the need for installing special IPSec-compatible stacks will be reduced.

FIGURE 5.1 IPv4 and IPv6 packet headers.

IPSec is built around a number of standardized cryptographic technologies to provide confidentiality, data integrity, and authentication. For example, IPSec uses the following:

- Diffie-Hellman key exchanges to deliver secret keys between peers on a public net
- Public-key cryptography for signing Diffie-Hellman exchanges to guarantee the identities of the two parties and avoid man-in-the-middle attacks
- DES and other bulk-encryption algorithms for encrypting data
- Keyed hash algorithms (HMAC, MD5, and SHA) for authenticating packets
- Digital certificates for validating public keys

The use of all these technologies within IPSec have been carefully laid out in architectural documents like RFC 1825 and newer versions (currently the latest Internet draft is draft-ietf-ipsec-arch-sec-05.txt). Figure 5.2 displays a conceptualization of the IPSec architecture, showing the relationships between the different components of IPSec. The three main components are the AH protocol, the ESP protocol, and key management. The design of the AH and ESP protocols are modular in nature, allowing different cryptographic algorithms to be used as desired. If new algorithms are developed, such as the elliptic curve algorithms that are now becoming commercially available, the parameters for their use can be standardized and then used in conjunction with AH or ESP.

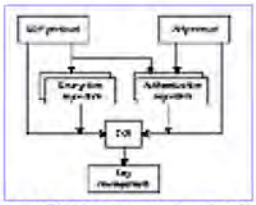


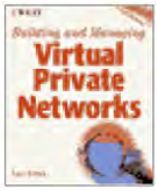
FIGURE 5.2 IPSec architecture.

Because the security services offered by IPSec use shared secret values (cryptographic keys), IPSec relies on a separate set of mechanisms for putting these keys in place.

When two parties want to exchange secure communications, they need to be sure that they're reading the same page in the playbook. The two parties have to be using the same cryptographic algorithm, the same key length, and the same keys if they're going to successfully exchange secure data; this is handled via a *Security Association* (SA). Although IPSec specifies default algorithms for authentication and encryption, it also allows for other algorithms to be used. To help simplify and organize many of the parameters that need to be specified for a Security Association, IPSec uses a *Domain of Interpretation* (DOI) to standardize the expected parameters for a given protocol's SA.

The Domain of Interpretation groups related protocols that are required for negotiation of a security association. Thus, a DOI includes information on a security protocol, its related cryptographic algorithms (such as DES, for example), and the requirements for exchanging keys to make that algorithm work properly. The DOI further sets out the format of any data, such as the key format, that should be transferred in an SA. It's much like deciding which language you and your correspondent are going to use for communicating via e-mail, but in this case, a DOI is designed for security associations.

Previous | Table of Contents | Next



Building and Managing Virtual Private Networks
by Dave Kosiur
Wiley Computer Publishing, John Wiley & Sons, Inc.
ISBN: 0471295264 **Pub Date:** 09/01/98

The Building Blocks of IPSec

Three main components are required for operating IPSec at its most basic level. These components are Security Associations, the Authentication Header, and the Encapsulating Security Payload.

Security Associations

Before we even get into the details of the protocols for authentication and encryption, we need to cover a very important concept in IPSec implementations, the Security Association. In order for two parties to exchange secured data (i.e., data that has been authenticated, encrypted, or both), both parties need to agree on which cryptographic algorithms they'll use, how to exchange keys, and then exchange the keys, if needed. They also may need to agree on how often they'll change the keys they're using.

All of these agreements have been bundled together in IPSec into a Security Association. Each secure communication between a sender and a receiver requires at least one SA and can require more than one because each IPSec protocol requires its own SA. Thus, authenticating a packet requires one SA, and encrypting that same packet requires another SA. Even if the same algorithms were used for authentication and encryption, two different SAs would be needed because two different sets of keys would be required.

A Security Association groups together all the things you need to know about how you communicate securely with someone else. An IPSec SA specifies the following:

- The mode of the authentication algorithm used in the AH and the keys to that authentication algorithm
- The ESP encryption algorithm mode and the keys to that encryption algorithm
- The presence and size of any cryptographic synchronization to be used in that encryption algorithm
- What protocol, algorithm, and key you use to authenticate your communications
- What protocol, encrypting algorithm, and key you use to make your communications private
- How often those keys are to be changed
- The authentication algorithm, mode, and transform for use in ESP plus the keys to be used by that algorithm
- The key lifetimes
- The lifetime of the SA itself
- The SA source address

You can think of the SA as your secure channel through the public network to a certain person, group of people, or network resource. It's like a contract with whomever is at the other end.

SAs are good for building multiple secure VPNs. Imagine that your company has its own VPN, and you develop a business relationship with another company that also has a secure VPN. You want to give them some access to your network by linking the two VPNs, but you don't want them to have full access to your network's resources. To accomplish this, you'd set up specific SAs between your VPN and theirs, controlling who has what access to which resources. And, you have a different set of specific SAs within your VPN for your employees, perhaps even broken down further by department.

SAs are good for only one-way communications—that is, they're defined for transferring data between a sender and a receiver but not for any exchanges in the opposite direction (i.e., from the sender back to the receiver). If two-way communications is necessary, two SAs must be agreed upon: one for data traveling from Ann to Tim, the other for data traveling from Tim to Ann.

The Authentication Header

In the IPSec system, a special header, the Authentication Header (AH), was designed to provide most of the authentication services for IP data. The AH contains a cryptographic checksum for the packet's contents. The Authentication Header is inserted into the packet between the IP header and any subsequent packets' contents (see Figure 5.3); no changes are made to the packet's data (the payload).

The Authentication Header contains five fields: the Next Header field found in all IP headers, a payload length, the Security Parameter Index, a sequence number, and authentication data. Two items are of particular note in the Authentication Header: first, the *Security Parameter Index* (SPI), which specifies to the device receiving the packet what group of security protocols the sender is using for communications; second, the authentication data itself, which is obtained by applying the cryptographic algorithm defined by the SPI to the packet's payload.

The new default methods for calculating the checksum are a relatively new cryptographic algorithm known as HMAC (for hash-based message authentication code) coupled with the MD5 hash function and HMAC coupled with the SHA-1 hash function. Both of these defaults are the result of recent changes to IPSec to improve the authentication mechanism, because the previous default, keyed MD5, was found to be susceptible to certain types of attacks called *collision attacks*, where a matching hash value is computed for two different messages.

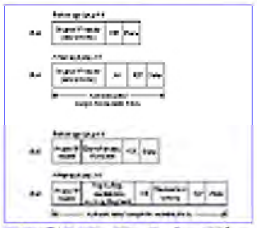


FIGURE 5.3 The Authentication Header.

The procedure for using either method (i.e., HMAC-MD5 or HMAC-SHA-1) is identical; SHA-1, however, is considered to be a stronger hash function than MD5. In both cases, the algorithm operates on 64-byte blocks of data. The HMAC-MD5 method produces a 128-bit authenticator value (or

cryptographic checksum), while HMAC-SHA-1 produces a 160-bit authenticator. Because the default authenticator length specified in AH is only 96 bits, either of the authenticator values produced must be truncated before storing the value in the authenticator field of the AH.

Upon receiving the packet, the recipient then would calculate his own 128-bit or 160-bit authenticator value (depending on whether HMAC-MD5 or HMAC-SHA-1 was used), truncate it according to the specified length of the authenticator field, and compare his authenticator value to the received authenticator value. As long as the two are identical, the data has not been altered in transmission.

AH in IPv4 versus IPv6

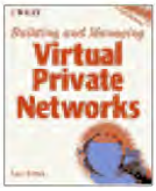
IPv6 is the next IP standard coming down the road. The IPv6 header is quite different from the existing IPv4 header. Among the more important changes, IPv6 headers carry 64-bit addresses instead of 32-bit addresses; IPv6 is expected to solve the problem of coming up with new IP addresses in an expanding Internet. It also is expected to enable a more flexible network architecture.

One of the difficulties with IPv6 headers and the host of optional headers that IPv6 specifies is that there is more in them that can change in transit through the network. This makes wrapping the AH's authentication around an IPv6 packet a little more complicated. However, the IPSec group has been developing AH in concert with IPv6 standards and has developed protocols for flexible ranges of authentication and intelligent placing of the AH in the IP packet so that it can work under either IPv4 or IPv6.

Because it's possible for an attacker to intercept a series of packets and then retransmit, or replay, them at a later time, AH also offers an antireplay service that can be invoked at the discretion of the receiver to help counter denial-of-service attacks that would be based on these retransmissions.

Note that the Authentication Header does nothing to keep the data confidential. If an attacker were to intercept the packets on the network, say with a sniffer, he still could read the contents of the packet, although he could not alter the packet's contents and resend the packets without changing the hash value. In order to protect the data against eavesdropping, we need to turn to the second component of IPSec, the Encapsulating Security Payload.

Previous | Table of Contents | Next



Building and Managing Virtual Private Networks
by Dave Kosiur
Wiley Computer Publishing, John Wiley & Sons, Inc.
ISBN: 0471295264 **Pub Date:** 09/01/98

ESP: The Encapsulating Security Payload

The second protocol in the IPSec scheme of things, the *Encapsulating Security Payload* (ESP), is responsible for encrypting a packet. Like the Authentication Header, the ESP header is inserted into the packet between the IP header and any subsequent packet contents (see Figure 5.4). However, because ESP is responsible for encrypting the data, the payload will be altered.

Like AH, the ESP header contains an SPI to indicate to the receiver what security association is appropriate for processing the packet. The sequence number found in the ESP header is a counter that increases each time a packet is sent to the same address using the same SPI. The sequence number indicates which packet is which and how many packets have been sent with the same group of parameters. The sequence number provides protection against replay attacks in which an attacker copies a packet and sends it out of sequence to confuse communicating nodes.

The remaining parts of the packet, except for the authentication data, are encrypted prior to transmission across the network. When unencrypted by the receiver, the new packet includes the payload data, up to 255 bytes of padding (to allow for the fact that certain types of encryption algorithms require the data to be a multiple of a certain number of bytes), and the pad length field, which specifies how much of the payload is padding as opposed to data.

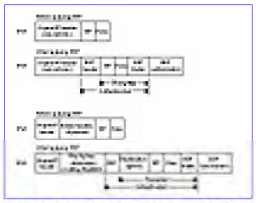


FIGURE 5.4 The ESP header.

ESP can support any number of encryption protocols; it's up to the user to decide which one to use. You can even use different protocols for each party with whom you're communicating. But, IPSec specifies a basic DES-CBC (DES with *Cipher Block Chaining*) cipher as its default, to guarantee a minimal interoperability among IPSec networks.

Using DES-CBC requires a 56-bit DES secret key, which is included as part of the security association. In order to use cipher block chaining, a 64-bit initialization vector is required, and the data is processed in 64-bit blocks; the packet's data is padded to create an integral number of 64-bit blocks if necessary.

ESP also can be used for authentication. The ESP authentication field, an optional field in the ESP header, contains a cryptographic checksum that's computed over the remaining part of the ESP (minus

the authentication field itself). This checksum varies in length depending on the authentication algorithm used. It may also be omitted entirely, if authentication services are not selected for the ESP. The authentication is calculated on the ESP packet when encryption is complete.

The current IPSec standard specifies HMAC with hash functions SHA-1 and MD5 as mandatory algorithms for IPSec-compliant equipment and software to support as authentication procedures in the ESP packet's authentication field.

The authentication provided by the AH differs from that provided in the ESP in that the ESP's authentication services do not protect the IP header that precedes the ESP, although they do protect an encapsulated IP header in tunneling mode (see the next section). The AH services protect this external IP header, along with the entire contents of the ESP packet (see Figure 5.5).

If AH was already designed for authenticating packets, why include an authentication option in ESP? AH is meant for occasions when only packet authentication is needed. On the other hand, when authentication and privacy are required, it's best to use ESP, including ESP's authentication option. Using ESP for encryption and authentication, rather than ESP and AH together, reduces the amount of copying done during packet processing and requires only one "transform" operation, rather than one each for ESP and AH, so packet processing is more efficient.

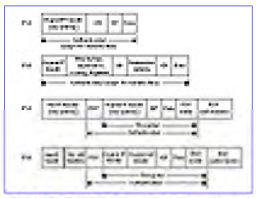


FIGURE 5.5 Authentication by AH versus authentication by ESP.

A Question of Mode

The IPSec specifications allow AH and ESP to be applied to an IP packet in two different ways, called modes. In *transport mode*, only the Transport-layer segment of an IP datagram is processed (i.e., authenticated or encrypted). The other approach, authenticating or encrypting the entire IP packet, is called *tunnel mode.*

Transport mode is applicable to either gateway or host implementations and provides protection for upper layer protocols, in addition to selected IP header fields.

In transport mode, AH is inserted after the IP header and before an upper layer protocol (e.g., TCP, UDP, or ICMP), or before any other IPSec headers that already have been inserted (see Figure 5.3), as described in the earlier section on AH. The IP address of the source and destination are still open to modification if the packets are intercepted.

In tunnel mode, the inner IP header contains the ultimate source and destination address, while the outer header contains other IP addresses (e.g., those of the security gateways). In tunnel mode, AH protects the entire inner IP packet, including the inner IP header (see Figure 5.6).

Because AH only protects the packet's contents against modification, other means are needed to ensure the data's privacy. In tunnel mode, the idea is to extend such protection to the IP header's contents, particularly the source and destination addresses. Although transport mode ESP is sufficient for

protecting the contents of a packet against eavesdropping, it does not provide total security for your traffic. A sophisticated attacker could still read the source and destination addresses of the packet and apply traffic analysis to learn of your communication patterns. If new correspondents were added, or traffic increased with a business partner, someone might learn something of value—for instance, that a merger was being planned or inventory increased for a new product rollout.

Tunnel-mode ESP provides more security for each IP packet by encrypting the entire packet (see Figure 5.7). After the packet's contents (including the original header) are encrypted, tunnel-mode ESP generates a new IP header for routing the secured datagram from sender to receiver.

Even tunnel-mode ESP does not guard against all types of traffic analysis on the Internet, because the IP addresses of the sending and receiving gateways can still be determined by examining the packet headers. This could enable an eavesdropper to learn that two different businesses are talking to each other or that traffic between them has increased, but it doesn't give the attacker any clue as to the persons within the two companies who are talking to each other.

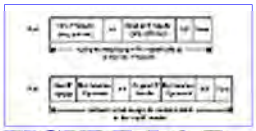


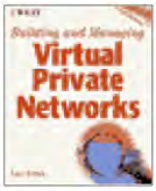
FIGURE 5.6 Tunnel-mode AH.

FIGURE 5.7 Tunnel-mode ESP.

In addition to applying either AH or ESP to an IP packet in transport or tunnel modes, IPSec requires support for certain combinations of tunnel and transport modes (see Figure 5.8). Basically, the idea is to use tunnel mode to authenticate or encrypt a packet and its header (IP1 or *inner header*), then apply AH, ESP, or both in transport mode to further protect the newly generated header (IP2 or the *outer header*).

Note that tunnel-mode applications have one less permutation than transport-mode applications: AH and ESP aren't used together in tunnel mode. The main reason for this is that ESP has its own authentication option. It's recommended that this option be used if a tunnel-mode packet needs both encryption and authentication of the inner packet.

Previous | Table of Contents | Next



Building and Managing Virtual Private Networks
by Dave Kosiur
Wiley Computer Publishing, John Wiley & Sons, Inc.
ISBN: 0471295264 **Pub Date:** 09/01/98

Key Management

With all the secret keys that have to be exchanged for different IPSec parties to communicate securely, it should be obvious that key management is an essential part of IPSec. Part of the procedure is handled by Security Associations and the SPI values that refer to them in each IPSec packet.

There are currently two ways to handle key exchange and management within IPSec's architecture: manual keying and *Internet Key Exchange* (IKE). Both of these methods are mandatory requirements of IPSec.

FIGURE 5.8 Transport and tunnel possibilities.

Key Exchanges Using SKIP

Another key distribution scheme, Simple Key Management for IP (SKIP), has been used by some companies for exchanging encryption keys. Instead of using session oriented keys, SKIP uses packet-oriented keys that are communicated in-line with the packets. Sun Microsystems, Novell and a few other companies currently offer security products that use SKIP for key distribution, but the IPSec Working Group has not pursued incorporating SKIP within its key management standards.

IPSec-compliant systems must support manual keying. In fact, for some time, this was the only way for vendors and other sites to exchange keys for interoperability testing. Face-to-face key exchanges, such as trading keys on paper or a floppy disk, can be used, or keys can be sent via a bonded courier or e-mail.

In 1996, RSA Data Security Inc., encouraged a group of product vendors to join together to test the interoperability of their IPSec products. This group, the S/WAN Initiative (S/WAN = Secure WAN), has run a number of interoperability trials since its formation. Prior to late 1997, almost all of the tests were run using the manual exchange of keys. To facilitate these exchanges, S/WAN published a recommended file format for the exchanges (see Figure 5.9).

Although manual keying is suitable for a small number of sites, scaleable, automated management is required to accommodate on-demand creation of SAs (e.g., for user- and session-oriented keying and to ease the use of the antireplay features of AH and ESP). The default automated key management protocol

for use with IPSec is IKE, which is the result of combining the *Internet Security Association and Key Management Protocol* (ISAKMP), which serves as a framework for authentication and key exchange, with the Oakley protocol, which describes various modes of key exchange. IKE is a relatively new name for ISAKMP/Oakley; you'll find many IPSec documents still refer to the key-exchange protocols as ISAKMP/Oakley. In this chapter, we'll use ISAKMP or Oakley as modifiers for concepts to emphasize the origin of a feature in IKE.

IKE is designed to provide four capabilities:

1. Provide the means for parties to agree on which protocols, algorithms, and keys to use.
2. Ensure from the beginning of the exchange that you're talking to the right person.
3. Manage those keys after they've been agreed upon.
4. Ensure that key exchanges are handled safely.

As you might expect, key exchange is closely related to the management of security associations. When you need to create an SA, you need to exchange keys. So IKE's structure wraps them together and delivers them as an integrated package.

FIGURE 5.9 Recommended file format for manual key exchange.

ISAKMP's Phases and Oakley's Modes

IKE operates in two phases, as originally defined in ISAKMP. In phase one, two ISAKMP peers establish a secure channel for performing ISAKMP operations (called the ISAKMP SA). In phase two, those two peers negotiate general purpose SAs.

NOTE: An ISAKMP peer is an IPSec-compliant node capable of establishing ISAKMP channels and negotiating SAs. It might be the computer on your desktop or a security gateway that negotiates security services for you.

Oakley provides three modes of exchanging keying information and setting up ISAKMP SAs: two for ISAKMP phase one exchanges and one for phase two exchanges.

1. *Main mode.* Accomplishes a phase one ISAKMP exchange by establishing a secure channel.
2. *Aggressive mode.* Is another way of accomplishing a phase one exchange. It's a little simpler and a little faster than main mode and does not provide identity protection for the negotiating nodes, because they must transmit their identities before having negotiated a secure channel.
3. *Quick mode.* Accomplishes a phase two exchange by negotiating an SA for general-purpose communications.

IKE also has one other mode, called new group mode, which doesn't really fit into phase one or phase two. The new group mode can only follow a phase one negotiation and is included to provide a mechanism for defining private groups for Diffie-Hellman exchanges.

NOTE: When preparing for a Diffie-Hellman exchange, certain material is needed to generate the keys;

this information is called a group and includes two numbers, a large known prime number and a seed.

To establish an IKE security association, the initiating node, a host or security gateway, proposes at least four items:

1. An encryption algorithm to protect data.
2. A hash algorithm to reduce data for signing.
3. An authentication method for signing the data.
4. Information about a group over which to do a Diffie-Hellman exchange.

A fifth item, a pseudo-random function used to hash certain values during the key exchange for verification purposes, also can be proposed in the Security Association. If it's not included, then the HMAC version of the hash algorithm specified in item 2 is used.

MAIN MODE

Main mode provides a mechanism for establishing the first phase ISAKMP SA, which is used to negotiate future communications. The steps are as follows:

1. Use main mode to bootstrap an ISAKMP SA for temporary communication.
2. Use quick mode within that ISAKMP SA to negotiate a general SA.
3. Use SA of step 2 to communicate from now until it expires.

The first step, securing an ISAKMP SA using Main mode, occurs in three two-way exchanges between the SA initiator and the recipient (see Figure 5.10). In the first exchange (steps 1 and 2 in the illustration), the two agree on basic algorithms and hashes. In the second exchange (steps 3 and 4), they exchange public keys for a Diffie-Hellman exchange and pass each other nonces—that is, random numbers that the other party must sign and return to prove their identity. In the third exchange (steps 5 and 6), they verify those identities and the exchange is completed.

In all of these steps, an ISAKMP header preceding the rest of the packet identifies the step being taken. Each of the items is carried in its own payload, but you can pack any number of these payloads into a single ISAKMP packet.

The parties actually use the shared key in three permutations, once they derive it. Both parties have to hash it three times: generating first a derivation key (to be used later for generating additional keys in Quick mode), then an authentication key, and, finally, the encryption key to be used for the ISAKMP SA.

Previous | Table of Contents | Next

A Main mode exchange protects the identities of the communicating parties. If it's not necessary to protect the identities, a faster exchange, the Aggressive mode, can be used.

FIGURE 5.10 ISAKMP Main mode.

AGGRESSIVE MODE

Aggressive mode provides the same services as Main mode, that is, it establishes the original ISAKMP SA. Aggressive mode looks much the same as Main mode except that it is accomplished in two exchanges, rather than three, with only one round trip, for a total of three packets rather than six.

In Aggressive mode, the proposing party generates a Diffie-Hellman pair at the beginning of the exchange and does as much as is practical with that first packet—proposing an SA, passing the Diffie-Hellman public value, sending a nonce for the other party to sign, and sending an ID packet that the responder can use to check their identity with a third party. The responder than sends back everything needed to complete the exchange; this response combines all three response steps in Main mode, so that the only thing the initiator has to do is confirm the exchange (see Figure 5.11).

Since the Aggressive mode does not provide identity protection for the communicating parties, it's necessary that the parties exchange identification information prior to establishing a secure SA in which to encrypt it. So someone monitoring an aggressive exchange can actually identify who has just formed a new SA. The advantage of Aggressive mode, however, is speed.

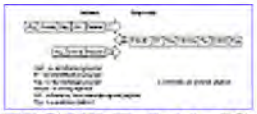


FIGURE 5.11 ISAKMP Aggressive mode.

QUICK MODE

After two communicating parties have established an ISAKMP SA using Aggressive mode or Main mode, they can use Quick mode.

Quick mode has two purposes: negotiating general IPSec security services and generating fresh keying material.

Quick mode is considerably simpler than either Main or Aggressive mode. Because it's already inside a secure tunnel (every packet is encrypted), it also can afford to be a little more flexible.

Quick mode packets are always encrypted and always start with a hash payload. The hash payload is composed using the agreed-upon pseudo-random function and the derived authentication key for the ISAKMP SA. The hash payload is used to authenticate the rest of the packet. Quick mode defines which parts of the packet are included in the hash.

Key refreshing can be done in one of two ways. If you don't want or need perfect forward secrecy (see Chapter 4, "Security: Threats and Solutions"), Quick mode can just refresh the keying material already generated in Main or Aggressive mode with additional hashing. The two communicating parties can exchange nonces through the secure channel and use these to hash the existing keys.

If you do want perfect forward secrecy, you can still request an additional Diffie-Hellman exchange through the existing SA and change the keys that way.

Basic Quick mode is a three-packet exchange, like Aggressive mode (see Figure 5.12).

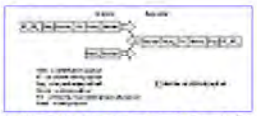


FIGURE 5.12 ISAKMP Quick mode.

If the parties do not require perfect forward secrecy, the initiator sends a packet with the Quick mode hash and a nonce. The respondent then replies with a similar packet but generating its own nonce and including the initiator's nonce in the Quick mode hash for confirmation. The initiator then sends back a confirming Quick mode hash of both nonces, completing the exchange. Finally, both parties perform a hash of a concatenation of the nonces, the SPI, and the protocol values from the ISAKMP header that initiated the exchange, using the derivation key as the key for the hash. The resulting hash becomes the new password for that SA.

If the parties do require perfect forward secrecy, the initiator first generates a public/private key pair and sends the public key along with the initiation packet (along with the hash and nonce). The recipient then responds with his or her own public key and nonce, and both parties then generate the shared key using a Diffie-Hellman exchange, again fully protected by the Quick mode hashes and by full encryption within the ISAKMP SA.

Negotiating the SA

Establishing a general purpose SA is relatively simple. To generate a new SA, the initiator sends a Quick mode message through the ISAKMP SA requesting the new SA. A single SA negotiation actually results in two SAs: one inbound, to the initiator, and one outbound. Each IPSec SA is one way; to avoid conflicting SPIs, the receiving node always chooses the SPI.

So, using Quick mode, the initiator tells the respondent which SPI to use in future communications with it, and the respondent follows up with its own selected SPI.

Each SPI, in concert with the destination IP address, uniquely identifies a single IPSec SA. However, in

practice, these SAs are always formed in pairs—inbound and outbound—and these pairs have identical parameters, keys, authentication and encryption algorithms, and hashes, apart from the SPI itself.

Using IPSec

Returning to our original schematic of an Internet VPN that was introduced in Chapter 3 (reproduced in Figure 5.13), it should be obvious that there are three major locations for installing IPSec-compatible software: on security gateways, mobile clients, and hosts on your corporate subnets.

Not all of the devices pictured need to have IPSec software in order to create an effective VPN—it depends on your needs. For instance, if you're looking to create only a LAN-to-LAN VPN, IPSec security gateways will suffice. On the other hand, if you have mobile workers and small branch offices that will need to dial into the corporate net via an ISP, then IPSec client software has to be installed on the appropriate computers—laptops for the mobile workers, perhaps the branch office's desktop computers. Lastly, if you want to create a VPN in which every computer can communicate with every other computer via IPSec protocols, then you'll have to deploy IPSec software on every host.

FIGURE 5.13 Components of an Internet VPN.

Security Gateways

We've already mentioned security gateways in describing VPNs. The security gateway is a network computing device, such as a router or firewall, that separates the internal, protected network from the external, unprotected network and performs cryptographic transforms on behalf of authorized users within the internal network.

Using IPSec on a security gateway means that the traffic of several hosts is funneled through a single encrypting host (i.e., the gateway) before it traverses the unprotected network. When constructing a VPN, you'd install a security gateway at each of your major offices and then establish security associations between each and every gateway.

Security gateways typically will establish and maintain individual security associations with each other. In other words, the gateways will use the same SA and related crypto keys regardless of whether Ann is communicating with Bill or Tim. If the three correspondents all had IPSec installed on their own computers, Ann would have to establish an SA to talk with each correspondent. Using security gateways reduces the complexity of key management, because only keys have to be assigned to the gateways. One fundamental requirement is that a network served by a security gateway should be internally secure (i.e., no "backdoor" unsecured entry points into the network exist, and it's understood that each individual is responsible for securing the data on his own computer).

Security gateways can transfer IPSec packets using either transport mode or tunnel mode. Selecting tunnel mode or transport mode for connections between your security gateways depends on your needs. For ultimate security, tunnel mode is preferred because it hides the IP addresses of the actual sender and receiver and guards against header cut-and-paste attacks. But, tunnel mode requires additional computation at the gateway and adds to the size of the packets, both of which can affect network

throughput. Furthermore, packets at one end of an IPSec tunnel must always be routed to the gateway at the other end of the tunnel. There is no mechanism to redirect a tunnel-mode packet if the gateway at the destination gets overloaded or crashes. But, if you set up security gateways to share SAs and the associated keys among one another, the IP routing can deliver the packets to a backup gateway.

Previous | Table of Contents | Next

Using transport mode between gateways reduces the communications overhead but does not hide the IP addresses of the ultimate source and destination. If a *wild card* security association isn't used for all traffic destined for a particular security gateway, then key management becomes more complicated. Without wild card SAs, Ann has to maintain an SA for each correspondent on the network served by the other gateway.

Wild Card SAs

Wild card security associations are used to simplify communications between hosts that are protected by security gateways. Rather than associate an SA with a specific host's IP address, the wild card SA is associated with all hosts on the LAN served by the security gateway.

When reviewing the features and capabilities of security gateways, such as encrypting routers, here are a few things you should look for:

- Support separate network connections for plaintext and ciphertext.
- Available key sizes must be consistent with the sensitivity of the information you'll transmit across the data link.
- If you decide that the default crypto algorithms will not meet your needs, the device should support the accepted alternative algorithms.
- Both AH and ESP should be supported.
- Manual input of SAs, including wild card SAs, should be supported.
- Mechanisms for protecting secret and private keys should be included.
- A system for changing crypto keys automatically and periodically makes key management easier and more secure.
- A security gateway should include some support for logging failures when processing a header; even better, some kind of alarm for persistent failures should be included.

Remote Hosts

When you're on the road or in a small branch office that uses dial-in connections to the corporate VPN, it's unlikely that you'd have a firewall or router installed on your computer to serve as a security gateway. Security gateways are meant to protect LANs, not individual computers. This means that IPSec-compliant client software has to be installed on your computer if you're going to connect to an IPSec-protected VPN. In most cases, this means that the TCP/IP stack running on your computer has to be modified to be IPSec-compliant, especially if you're running IPv4. (Recall that IPSec is an add-on

feature for IPv4, although it's an integral part of IPv6.)

Header Cut-and-Paste Attacks

If packets are encrypted, but not authenticated, an attacker can make copies of all packets transmitted between two parties and use those packets to forge a message or eavesdrop on an encrypted one. The attacker can copy the encrypted message from the original packets and send it to another correspondent with a new packet header. This correspondent (or co-conspirator, if you will) can decrypt the message as long as it's routed through the same security gateway.

In IPv4 implementations, IPSec support can be inserted in one of two locations in the TCP/IP stack. In one case, the IPSec code can be inserted between the network and transport layers. In the second case, the necessary code can be inserted as a *shim* between the Data Link layer and the Network layer. The first case offers users more flexibility because it enables them to assign different security associations for different software; in other words, some traffic could be transmitted without IPSec because it's not needed, while other, more important traffic could be set to be transmitted with IPSec security. The shim approach can be easier to implement, but it can enforce security associations only at the IP address level and cannot enforce user identities.

One concern with handling remote client access is how to distribute the needed security associations. A practical approach is to have a central site generate all SA parameters and then send them to the clients, perhaps using the S/WAN format.

Another potential problem is handling the IP addresses for the remote clients. Because many mobile clients are likely to dial into the VPN via their local ISP, they'll often be assigned a variable IP address that's only good for that connection. Thus, the client's SA with the central site has to be able to work with a variety of IP addresses, some of which might not be known ahead of time. One solution is for the client not to make assumptions about its local address and to use a wild-card specification of central site addresses.

Just as we listed requirements for security gateways, here are some features to check when evaluating client software:

- Compatibility with other IPSec implementations; for example, match the site's encrypting server (transport and tunnel modes, key exchange protocol, crypto algorithms, etc.).
- Offers a clear indication of when IPSec is working.
- Supports downloading SAs (via paper or disks, for instance).
- Has to handle dynamically assigned IP addresses.
- Includes mechanisms to protect the keying material from theft (encrypt keys with passwords, for instance).
- Offers a mechanism to change the crypto key automatically and periodically; includes dynamic assignment of new SPI numbers during rekeying; compatible with standard IPSec keying protocols; uses a cryptographically strong random-key procedure to generate its keys.
- Explicitly blocks non-IPSec traffic.

A large number of vendors already support IPSec (see Table 5.1 for a partial list). If you're already working with an installed base of network equipment that can be upgraded to support IPSec, but not all

the equipment is from one vendor, be sure to check on the product's IPSec interoperability. The IPSec Working Group has published a series of suggested tests for IPSec interoperability (see web.mit.edu/tytso/www/ipsec/companies.html for details on vendor implementations and interoperability results). The *Automotive Network Exchange* (ANX) also has been pushing the implementation of IPSec and has been running its own interoperability tests (see www.aiag.org/anx/). Many products are interoperable at the level of AH and ESP headers but may not support all key management features, for example.

Previous | Table of Contents | Next

TABLE 5.1 Partial List of IPSec Products

Vendor	*Product*
3COM	Secure VPN/NetBuilder
Ascend Communications, Inc.	Secure Access
Bay Networks (New Oak)	NOC 4000 Extranet Access Switch
Bellcore	ERP IPSec
Cabletron/Network Express	NE-Secure
Check Point Software Technologies	Firewall-1
Cisco Systems	Cisco IOS
ftp Software	OnNet
Frontier Technologies Corp.	e-Lock VPN
Gemini Computers Inc.	Trusted Security Firewall-Guard
IBM	IBM SNG
Information Resources Engineering, Inc.	SafeNet
Lucent (Livingston Enterprises)	Livingston ComOS
Mentat Inc.	Mentat TCP
Network Associates, Inc.	Gauntlet
Network Systems	BorderGuard and Security Router
Raptor Systems	Eagle VPN
Secure Computing Corp.	BorderWare Firewall Server
TimeStep Corp.	PERMIT/Gate
TimeStep Corp.	PERMIT/Client
Toshiba Corp.	Network CryptoGate
V-One V-ONE	SmartWall
VPNet Technologies Inc.	VSU-1000 VPN Service Unit

In mid 1998, the International Computer Security Association (ICSA) also started certifying products for compliance with the IPSec protocol specifications. Check their Web site at www.icsa.net for more details.

Other details of IPSec software and hardware are presented in Chapters 10 through 12.

Tying It All Together

Admittedly, full-fledged implementations of IPSec involve a lot of different interactions: matching the security associations and keys to corporate security policies; negotiating security associations and exchanging keys between gateways and/or hosts; and the very important end-results of authenticating and encrypting the packets. Although IPSec itself doesn't define how security policies are to be formulated and distributed, many current industry efforts aim at using X.500 or LDAP-compatible directories to store security policies for users and/or devices. Figure 5.14 schematically represents how security policies fit into the rest of the IPSec operation for a session between two hosts.

Sample Deployment

To illustrate the use of IPSec in a corporate VPN, let's create a relatively simple VPN (see Figure 5.15) that's composed of two sites: the corporate headquarters and a regional office. Mobile workers also are given the capability to dial into the VPN via local ISPs. We'll use encrypting routers as the security gateways.

Traffic inside the corporate networks is transmitted as plaintext and would be protected from outside attackers with techniques other than IPSec, such as firewalls, access control lists on servers, and so on). Only traffic between sites, or between mobile workers and a main site, is protected with IPSec. This VPN design is likely to be one of the more common designs.

To secure this system, physical security should include ensuring that all hosts reside within the site's physical parameters and that all links to outside systems go through the encrypting routers. The connection between the site's internal networks and the external network(s) should be in a locked machine room with restricted access, and only authorized individuals (network managers, for instance) should have access to the encrypting routers.

Key assignment and management for the main sites should be fairly straightforward because only two static sites are involved, the main headquarters and the regional office. Both sites require a security association with each other's encrypting router, but that's all that is needed for users at either site to communicate with the other site.

FIGURE 5.14 IPSec and security policies.

If the number of static sites grows, then it's probably best to use a central location for assigning SAs and keys. A hub-and-spoke topology may be needed in larger organizations as the relationships between different sites get more complicated. For example, each regional office might receive its SAs from the corporate headquarters, but a regional office could be responsible for issuing keys and SAs for the manufacturing plants or branch offices in its area. Some plants might also communicate with each other

frequently, but not with other offices, so they could choose to set up an SA between themselves without the knowledge of the regional office.

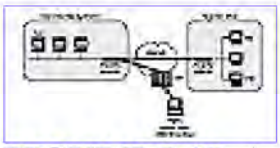
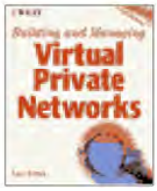


FIGURE 5.15 An example IPSec VPN.

In our example, key management for the mobile workers might demand the most attention. In this case, deciding between a centralized key-management system versus a distributed one depends on the number, and needs, of the mobile workers. If all the mobile workers only need to connect to either the corporate headquarters or the regional office, then that site should be responsible for issuing keys and SAs. In most companies, it's highly unlikely that many workers would need dial-in access to more than one site; in such cases, the keys probably should be assigned by what they call their "home office" and the appropriate keys and SAs disseminated to the other sites as needed.

Previous | Table of Contents | Next



Building and Managing Virtual Private Networks
by Dave Kosiur
Wiley Computer Publishing, John Wiley & Sons, Inc.
ISBN: 0471295264 **Pub Date:** 09/01/98

Remaining Problems with IPSec

Although IPSec already incorporates many of the necessary features for deploying secure VPNs over the Internet, it's still a work in progress. Here's a brief rundown of some of the issues that may affect your deployment of IPSec. None of these are likely to be show-stoppers—keep in mind that various working groups in the IETF are working to solve many of these problems.

All IP packets processed with IPSec increase in size due to the addition of IPSec headers, which may lead to increased packet fragmentation and reduced throughput. It's been proposed that this problem can be addressed by compressing the packet's contents before encryption, but this has not been standardized yet. VPNet already offers compression in its IPSec hardware. Also, the overhead associated with key-management protocols like IKE will reduce the available bandwidth on a link.

IKE is still a relatively unproven technology. For example, much of the original interoperability work performed under the aegis of S/WAN in 1996 and 1997 used manual keying. This may not prove to be a problem if you're focusing on a limited number of security gateways, but manual keying is not a suitable procedure for handling host-based IPSec or large numbers of mobile workers.

Remember that IPSec is designed to handle IP traffic only; it cannot transform IPX, AppleTalk, or NETBEUI traffic. If you're running a multiprotocol network, you may have to deploy one of the other protocols that we'll be describing in subsequent chapters. Alternatively, you could plan to migrate your network just to TCP/IP protocols. Products like Novell's Netware have IP gateways and are being migrated to native support of IP as well.

The computational overhead associated with many of the cryptographic algorithms used in IPSec can still pose problems for older workstations and PCs, so deployment of IPSec at the desktop level can affect performance considerably (see Figure 5.16).

Distribution of cryptographic software and hardware is still subject to government restrictions (and not just in the United States). These restrictions may require additional management duties if you're running an international organization, because you'll have to set up one set of SAs and keys for use within the United States and at least one more set for your international branches.

Encrypting the original packets can cause problems if the network is to provide differentiated service classes because the pertinent data will be encrypted and hidden from routers on the path between the source and receiver. If service classes need to be implemented, host-level security rather than gateway-level security should be implemented. This is also less of a problem if IPv6 is used, because IPv6 headers include information for class of service that would not be encrypted.

Lastly, using IPSec for tunneling allows nodes to have illegal IP addresses (that is, ones meant only for internal use) and still communicate with each other. But, if you should switch to host-level security, the IP addresses for your various corporate subnets will have to be more carefully managed to ensure that they comply with each other.

Summary

This chapter dealt with many of the details of the IETF's favored system for creating VPNs and securing data over the Internet—IPSec. The system includes a great deal of flexibility in authentication and encryption algorithms, allowing it to meet the demands of both current and future networking situations. The AH and ESP protocols can be applied either to authenticate and/or encrypt just the packet's payload (transport mode) or the entire IP header, including the IP addresses of the source and destination, as well (tunnel mode). The greatest degree of security is provided by applying authentication and encryption in tunnel mode.

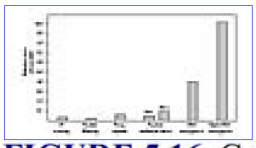
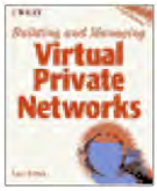


FIGURE 5.16 Computational cost per function.

In order to enable secure communications between two parties, a system for exchanging keys is required. IPSec's *Security Associations* (SAs) are created between correspondents to exchange keys as well as any pertinent details on the cryptographic algorithms that will be used for a session. Although manual exchanges of SAs and keys are possible for a small number of correspondents (or VPN sites), IPSec includes a fairly involved, but workable, framework for automatic key management called *Internet Key Exchange* (IKE) or ISAKMP/Oakley.

IPSec software can reside in stationary hosts, mobile clients, or security gateways. Only security gateways are needed if LAN-to-LAN tunnels need to be created. Mobile workers also would require IPSec client software if they wanted to connect to a VPN site. Should you want to maintain the identity of each correspondent (say for class-of-service differentiation), then installing IPSec on each and every computer may be necessary.

Previous | Table of Contents | Next



Building and Managing Virtual Private Networks
by Dave Kosiur
Wiley Computer Publishing, John Wiley & Sons, Inc.
ISBN: 0471295264 **Pub Date:** 09/01/98

CHAPTER 6
Using PPTP to Build a VPN

If you managed to work your way through Chapter 5, you now have an idea of just how complicated a Virtual Private Network can become. For someone planning an international VPN that provides the best security possible, supports both LAN-to-LAN tunnels and client-to-LAN tunnels, and still complies with national restrictions on cryptographic algorithms, IPSec's flexibility and numerous options are a must.

But, not all VPNs need to be so involved. Smaller businesses may only need a local or regional VPN with a limited geographic span. Some businesses may be more interested in supporting only their mobile workers and telecommuters with remote access via a VPN. These businesses could also benefit from using IPSec for their VPNs, but the market fragmentation has led to other solutions, such as PPTP (Point-to-Point Tunneling Protocol) and L2TP (Layer2 Tunneling Protocol), which are simpler and do not offer all the options, or protection, that IPSec does.

For those of you who have been following the VPN market in some detail, it may seem somewhat artificial to devote separate chapters to PPTP and L2TP and then to say nothing about L2F (Layer2 Forwarding). Originally, two simple tunneling protocols were proposed: PPTP by Ascend and Microsoft and L2F by Cisco. Because Microsoft has been actively supporting PPTP in its Windows NT Server (versions 4.0 and above) and because Ascend Communications (and other vendors) now include support for PPTP in hardware used by many ISPs, PPTP has become a popular method for constructing simple VPNs. On the other hand, L2F has stayed primarily a proprietary product from Cisco, and some of its features are being incorporated into L2TP. Because Microsoft's early support of PPTP makes it a current popular choice for VPNs, it should be covered in detail on its own, even if it will be superseded by L2TP, which is the plan of Microsoft and other vendors. L2F doesn't benefit from the same popularity, but its successor, L2TP, is likely to do so.

This chapter starts out with an overview of the architecture of PPTP and moves on to the details of how the protocol works. Then we move on to an overview of the types of products you can use to build a VPN using PPTP.

What Is PPTP?

The Point-to-Point Tunneling Protocol was first created by a group of companies calling themselves the *PPTP Forum*. The group consisted of 3Com, Ascend Communications, Microsoft, ECI Telematics, and US Robotics. The basic idea behind PPTP was to split up the functions of remote access in such a way

that individuals and corporations could take advantage of the Internet's infrastructure to provide secure connectivity between remote clients and private networks. Remote users would just dial into the local number of their Internet Service Provider and could securely tunnel into their corporate network.

The most commonly used protocol for dial-up access to the Internet is the *Point-to-Point Protocol* (PPP). PPTP builds on the functionality of PPP to provide dial-up access that can be tunneled through the Internet to a destination site. As currently implemented, PPTP encapsulates PPP packets using a modified version of the *Generic Routing Encapsulation* (GRE) protocol (see Figure 6.1), which gives PPTP the flexibility of handling protocols other than IP, such as IPX and NETBEUI, for example.

Because of its dependence on PPP, PPTP relies on the authentication mechanisms within PPP, namely PAP and CHAP; since there's a strong tie between PPTP and Windows NT, an enhanced version of CHAP, MS-CHAP, is also used. This version utilizes information within NT domains for security. Similarly, PPTP can use PPP to encrypt data, but Microsoft also has incorporated a stronger encryption method, *Microsoft Point-to-Point Encryption* (MPPE) for use with PPTP.

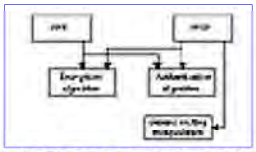


FIGURE 6.1 PPTP's architecture.

Aside from the relative simplicity of client support for PPTP, one of the protocol's main advantages is that PPTP is designed to run at Layer2, or the Link layer, as opposed to IPSec, which runs at Layer3. By supporting data communications at Layer2, PPTP can transmit protocols other than IP over its tunnels. IPSec, on the other hand, is restricted to transferring only IP packets over its tunnels.

Microsoft's inclusion of support for PPTP in its Windows NT Server and offering free clients for certain *Operating Systems* (OSs)—for NT and Windows95, for example—has made PPTP a popular method for creating dial-in VPNs. Microsoft's implementation of PPTP may not be a standard that's been ratified by a standards body like the IETF, and it may not even achieve the status of a *de facto* standard for VPNs due to its succession by L2TP. But, considering that so many of PPTP's features are tied to Windows NT and that Microsoft has tremendous influence in the PC world, it shouldn't come as a surprise that many of the initial products for PPTP have followed Microsoft's feature set. In fact, if your company is primarily a Windows shop, then setting up and using PPTP is fairly simple.

Because the *de facto* PPTP implementation is the one compatible with Microsoft's Windows NT version, this description of PPTP focuses on that implementation. As we go along, we'll note where the protocols and implementations differ from IETF standards or other documents that have been submitted to the IETF for consideration as standards.

Development of PPTP has proceeded in a number of different directions, leading to differing functionality among current and planned products. This means that you should exercise extra caution when selecting products and planning to use PPTP, because some products may not include the features you're planning for your VPN. For instance, Microsoft has included PPTP support in Windows NT 4.0 and released a *Dial-Up Networking Pack* (DUN) for Windows95 that includes PPTP, but these products support only client-to-LAN tunneling. LAN-to-LAN tunneling was introduced with the release of the *Routing and Remote Access Server* (RRAS) for Windows NT 4.0 in late 1997 and is planned for