

Privacy Issues in Virtual Private Networks

W. Timothy Strayer

BBN Technologies
10 Moulton Street, Cambridge, MA 02138

strayer@bbn.com

ABSTRACT

The term “private” in virtual private network is sometimes taken for granted, and people can be too distracted by the “virtual” and “network” to really consider what is meant by “private.” In this paper, we consider the issues and meanings of the term private, and look at these issues with respect to two of the dominant VPN technologies, IPsec and MPLS.

1 Introduction

As often happens with fairly simple concepts that become immensely popular, the term virtual private networks—VPNs—has come to mean many things to many people. There are dozens of books, hundreds of product sheets, and thousands of articles that, somewhere in the first several paragraphs, begin a sentence, “A *virtual private network* is a ...” and go on to define it in whichever way serves the context. Along the way, the term picks up so much baggage that it’s hard to find an actual virtual private network in a lot of proposed VPN solutions. The interesting thing is that the concept of a VPN is in fact very simple. To prove it, and to keep with tradition, this paper will make the following statement:

A virtual private network is a private network running over a shared public infrastructure like the Internet.

At its core, a VPN is a private network used to interconnect various geographically separated sites, or connect remote users back to a home network, or even to allow controlled access between different corporate networks. What makes this interesting is that the private network is actually constructed from protocols and technologies that run over a shared network. Rather than paying the subscriber fees to have leased lines forming a real private network, a VPN attempts to provide the same characteristics as leased lines but with the cost efficiency and flexibility of using the Internet. Indeed, the three arguments most often seen supporting VPNs is that they are less costly, more flexible, and easier to reconfigure than the traditional leased line approach.

VPNs are *constructed* private networks in the sense that it takes an integrated set of various technologies to provide what looks like a private network over a shared one. These technologies include

Views and conclusions contained in this paper are those of the author and should not be interpreted as representing BBN official policies, either expressed or implied.

A version of this paper appears in *Computer Communications*, Volume 27, Issue 6, April 2004, Pages 517-521.

- A tunneling protocol, like IPsec, PPTP, L2TP, or MPLS
- An authentication mechanism, as provided by PKI, RADIUS, or Smartcards
- An access control mechanism, as provided by Directory Servers and ACLs
- Data security technologies, like encryption
- Data provisioning techniques, like QoS and traffic engineering

When building a VPN solution, one generally picks a technology from each of the categories above. The decision about which technologies to use is ideally based on the requirements of the situation. In reality, the VPN equipment or service provider has often made some or all of these decisions already, and the buyer is left with a full package but little choice. That’s okay as long as the buyer is at least aware of the requirements, and can differentiate between VPN equipment or services based on what is appropriate for the buyer’s situation, not marketing hype and confusing product sheets.

In this paper we look at one aspect of VPNs, what we consider the core issue—privacy. It turns out that there are two different but equally valid definitions for privacy and, consequently, two philosophies for building VPNs based on technologies that support one or the other definition.

One way to consider privacy is to define it colloquially as “no one can see your stuff.” The emphasis here is on security, using techniques such as cryptography to achieve confidentiality. It is assumed that some aspect of the communications can be witnessed by others, so appropriate measures are taken to obscure the content and, sometimes, the nature of the conversation.

The other way to consider privacy is with respect to availability. Here, part of a shared infrastructure is carved out and dedicated only to your communication, and no one else can use your part when you are using it. This type of privacy requires provisioning and managing the resources in the network.

One of the most important and fundamental technology choices one makes before constructing a VPN is which tunneling protocol to use. In the next section, we give a quick overview about what tunneling is and is trying to accomplish, then we talk in more detail about two tunneling protocols, IPsec and MPLS, and how these two protocols, when used as the basis for VPNs, serve as embodiments of the two philosophies on privacy.

2 Tunneling

Tunneling protocols provide a way to overlay a virtual network over a physical one by building tunnels, or special connections, between various points in the physical network. The idea of tunneling is not new—it has been a very useful trick for picking up a packet running on one network, packaging it inside another packet, sending that package over to another network, unpacking the original packet, and letting it continue its journey.

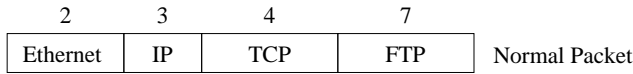


Figure 1: A “normal” packet showing all the layer numbers.

A typical network protocol stack starts with application data and forms a packet—the atomic unit of the Internet—by breaking up the application data into chunks, surrounding the chunks with a header (and sometimes a trailer) of the transport and network layer protocols. The packet contains the addressing information to carry the data from the source of the data to its final destination. This packet is then placed inside a link layer frame. The frame is only used for carrying the packet from one router to the next, or one “hop” along the path made up of many hops. This “normal” packet structure is shown in Figure 1.

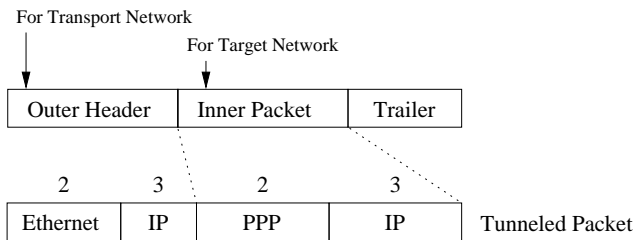


Figure 2: A tunneled packet showing how the layer numbers can be repeated.

Tunneling repeats some of the layers to play some tricks with how the application data eventually arrives at the destination. Consider an intranet running a private addressing scheme which is split over two sites, as shown in Figure 3. The private addresses are only meaningful within the intranet but not in the Internet—that is, a packet constructed from within the intranet cannot be routed through the Internet. A tunnel can be constructed from one site to the other such that the packet, carrying the private addresses, is placed inside the tunnel and transported to the other site, where the private addresses are again meaningful. This is done by constructing an *encapsulating* packet that takes the original packet as payload, as shown in Figure 2. The encapsulating packet travels from one site to the other through the Internet using addresses that the Internet understands. This way the original packet is never exposed to the Internet, so the private addresses are no longer an issue.

Notice that Figure 2 also shows the way the layers are repeated when one packet becomes the payload of another. This is an interesting side effect of tunneling—layers can become inverted in strange ways. Some situations call for layer 3 tunnels to transport other layer 3 packets, so layer 3 headers are repeated: outer layers

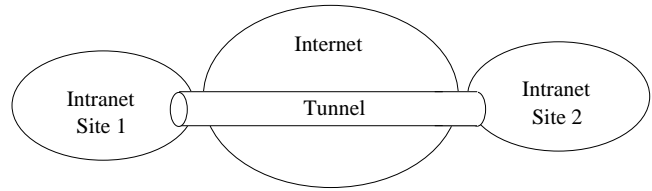


Figure 3: A tunnel between privately addressed intranet sites.

2 and 3 carrying inner layers 3, 4, and 7. Sometimes layer 2 frames are transported by TCP, so the layers look like outer layers 2, 3, and 4 carrying inner layers 2, 3, 4, and 7.

Tunneling can happen at virtually any layer, but the two layers where tunneling occurs most often are layer 2—the medium access control (MAC) layer—and layer 3—the network layer. Layer 2 tunneling, especially with respect to VPNs, generally happens when PPP, the Point-to-Point Protocol used by dial-in services, is tunneled through the Internet for part of the journey back to the home network, instead of using the phone lines the whole way. Three popular protocols for building layer 2 VPNs are the Point-to-Point Tunneling Protocol (PPTP) [2] developed by Microsoft and others, Layer 2 Forwarding (L2F) [7] developed by Cisco and others, and the Layer 2 Tunneling Protocol (L2TP) [6] developed by the IETF as a compromise between PPTP and L2F. For this paper, however, we’re going to discuss layer 3 VPN tunneling protocols because they provide site-to-site solutions.

Tunneling is a very useful tool, and preserving private addressing is only one of the reasons someone might use a tunnel. There are two other reasons of particular interest in VPNs—tunnels for data security, and tunnels for quality of service. VPNs based on the IPsec tunneling protocol are designed to satisfy the users interested in keeping their data secret, and VPNs based on MPLS are designed to satisfy users looking for provisioned levels of service.

Let’s look at these two protocols more closely.

2.1 IPsec

IPsec [5] is a suite of protocols—specified by RFCs 2401 through 2412—that defines the architecture for providing security services within the IP protocol. IPsec is designed to provide interoperable, strong cryptographically-based security services such as access control, data integrity, data origin authentication, anti-replay protection, and data confidentiality.

At the core, IPsec is built on two security protocols, the Authentication Header (AH) protocol [3], and the Encapsulating Security Protocol (ESP) [4]. Given an IP packet, the AH protocol places a new header just after the original IP header. This header carries information used to confirm origin authentication and data integrity, but it does not provide for encryption of the original packet (data confidentiality). ESP additionally provides data confidentiality by encrypting the original packet. Both protocols are offered in two modes, *transport mode* and *tunnel mode*. Transport mode is designed primarily to protect the higher-layer protocols such as TCP and UDP, and uses the original packet’s header to steer the IPsec packet. In tunnel mode, the original packet is encapsulated inside a new IPsec packet that also carries all of the security information

that IPsec needs to perform AH or ESP.

Since ESP provides a more complete set of security features, and the tunnel mode satisfies the need to create an overlay network on top of the public one, VPNs based on IPsec use ESP in tunnel mode exclusively, and not AH or transport mode.

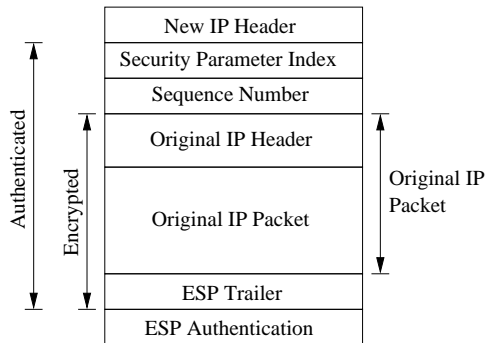


Figure 4: IPsec packet with ESP in tunnel mode.

An IPsec packet with ESP in tunnel mode is shown in Figure 4.

2.2 MPLS

The original motivation for MPLS—Multi-Protocol Label Switching [1]—was to enable routers to decide very quickly how to move packets on to their destination. The inspiration came from Asynchronous Transfer Mode (ATM). ATM has a similar mechanism called the VCI/VPI pair (virtual circuit identifier/virtual path identifier). The idea is to separate the routing function from the forwarding function within a router. Routes are set up through the network *a priori*. A route, then, can be defined as a series of VCI/VPI values. To forward an ATM cell (the moral equivalent of an IP packet—ATM’s atomic unit for data transport), the ATM switch simply looks up the VCI/VPI number in the cell, finds a second VCI/VPI value in its tables, replaces the cell’s VCI/VPI pair with the pair from the tables, and sends the cell out the link associated with that new VCI/VPI pair. At the next ATM switch, the same process happens, until the cell arrives at its final destination.

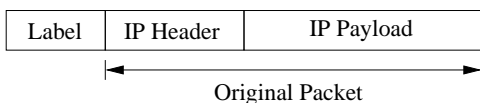


Figure 5: An MPLS “packet” with label attached.

MPLS works exactly the same way, except that the VCI/VPI pairs are now called *labels* that attach to the front of packets, as shown in Figure 5. As it became increasingly clear that ATM was not going to replace IP networks anytime soon, some of the good ideas from ATM started migrating into IP networks. Routing IP packets takes a fair amount of time and computation because the route of each packet is calculated as that packet proceeds through the network. Some people saw ATM’s switching paradigm as a way to reduce that overhead, especially if the routes can be calculated once beforehand. So some routers were modified to look for IP packets with special labels placed on the front, and handle them by doing

a simple table lookup rather than a full route calculation. This is called *IP cut-through routing*.

An interesting result of this is that these “label switch routers” (as they were called) only need to see the label, they don’t care about what follows the label, hence the “multi-protocol” part of the MPLS name.

The need-for-speed argument justifying MPLS was rather short-lived, however, due to new advances marrying switching technology with traditional IP routing. Undaunted, the MPLS advocates suggested that, even if speed was no longer the driving forces, the labels facilitated the same kind of traffic engineering and quality of service (QoS) that made ATM so attractive. Constructing a route through a network by first loading the label switch routers with the appropriate labels is tantamount to setting up connections, which means that the routers can keep per-route state. Traditional IP routers expressly didn’t keep state per route, and so were at a disadvantage with respect to providing QoS.

3 Head-to-Head Comparison

Why choose MPLS or IPsec as the basis for a VPN? In general, MPLS-based VPNs seek to establish tunnels that guarantee certain levels of service. The labels represent routes through the network, and those routes, when coupled with resource management mechanisms, can provide the users of the VPN with a service provisioned for their needs. Unfortunately, the structure of the Internet isn’t by its nature ready to support unrestricted MPLS, much less be able to make service level guarantees. The Internet is comprised of many Autonomous Systems (AS’s), each run by an Internet Service Provider (ISP). MPLS works well within a single AS, but it doesn’t traverse the AS boundaries because the labels are not meaningful outside of an ISP’s domain of control. Even so, the ISPs are deploying MPLS-equipped label switched routers within their own fairly extensive networks. ISPs have recognized this as a selling point, and many offer a premium provisioned services with service level agreements (SLAs) codifying the guarantees. But since labels are only meaningful within the ISP’s network, the provisioned service must begin and end within the ISP’s borders. This tends to keep an ISP’s customers from wandering to other service providers.

IPsec-based VPNs, in general, are concerned with establishing tunnels with very strong security mechanisms to effect the privacy of the VPN. Data carried by conventional IP packets through the Internet is completely exposed to anyone with the desire and a modest amount of equipment. IPsec’s authentication and confidentiality functions are important to businesses and other groups with data they wish to keep from this kind of exposure. But since IPsec packets are in fact just regular IP packets when viewed by a router, transporting IPsec packets does not require any special mechanisms or modifications of the routers in the Internet. This also means that IPsec-based services are not ISP dependent.

The differences between IPsec and MPLS with respect to VPNs really comes down to a dichotomy of assumptions. IPsec assumes that the goal is not to trust any intermediate systems. This feathers nicely with the general goal of IP, where the failure of any set of routers is recoverable because no router has to maintain state. The consequence, then, is that IPsec packets are at the mercy of the Internet, with no better or worse treatment than any other IP packet. MPLS, on the other hand, assumes that entities within the network

can be trusted to help achieve an engineered delivery.

4 Conclusion

We have seen that VPNs based on IPsec or MPLS approach privacy in virtual private networks with two different philosophies. One focuses on security, attempting to satisfying the need to keep data private. The other uses resource provisioning and traffic engineering to ensure that the network, while built of public parts, appears to the users as if it were a dedicated private network.

Which is the right way? That depends on two things: What control you are willing to cede to service providers, and what you mean by private. If you are looking for a VPN that is a black-box service with an SLA that has 5 or 6 nines in the availability column, and you are able to stay completely within one ISP's network, that suggests one solution over the other. Likewise, if it really matters to you that your data is kept strictly confidential, and that no one other than authorized recipients has the capability to read that data, then that suggests the other solution.

As these things go, however, the solution space does not consists simply of one or the other. Viewing IPsec and MPLS as having strengths and weaknesses, a trend now in VPN deployment is to use IPsec for the security aspects (what IPsec does best), and tunnel that through MPLS for quality of service (what MPLS does best).

Of course, building a real VPN involves many more decisions than just which tunneling protocol to use—how authentication is provided, how access is controlled, how data is kept confidential, how data integrity is maintained, how the whole system is managed, among other questions that lead to a complete and deployable solution.

Virtual private networks is a fairly simple concept. The set of technologies that must be pieced together to form a solution appropriate for a given deployment scenario is complicated. It is a failing of English, however, that the terms “virtual,” “private,” and “network” do not have a single binding—that is, are we talking about a “virtual private network” or a “virtual private network”? The first grouping makes sense and is certainly what the designers of VPNs have in mind, but with so many choices and so many purveyors of solutions, unless one is an educated consumer, one is very likely to end up with a solution that is only virtually private, no matter the definition of private.

References

- [1] E. ROSEN, VISWANATHAN, A., AND CALLON, R. Multiprotocol label switching architecture. RFC 3031, IETF, Jan. 2001.
- [2] HAMZEH, K., PALL, G., VERTHEIN, W., TAARUD, J., LITTLE, W. A., AND ZORN, G. Point-to-point tunneling protocol (L2TP). RFC 2637, IETF, July 1999.
- [3] KENT, S. T., AND ATKINSON, R. IP authentication header. RFC 2402, IETF, Nov. 1998.
- [4] KENT, S. T., AND ATKINSON, R. IP encapsulating security payload (ESP). RFC 2406, IETF, Nov. 1998.
- [5] KENT, S. T., AND ATKINSON, R. Security architecture for the internet protocol. RFC 2401, IETF, Nov. 1998.
- [6] TOWNSLEY, W. A., VALENCIA, A., RUBENS, A., PALL, G., AND PALTER, G. Z. B. Layer two tunneling protocol ‘L2TP’. RFC 2661, IETF, Aug. 1999.
- [7] VALENCIA, A., LITTLEWOOD, M., AND KOLAR, T. Cisco layer two forwarding (protocol) ‘L2F’. RFC 2341, IETF, Apr. 1998.