Understanding Internet Protocol Security

J Thomas and AJ Elbirt
Electrical and Computer Engineering Department
University of Massachusetts Lowell
One University Avenue
Lowell, MA 01854, USA

1 Introduction

The Internet Protocol Security (IPsec) architecture is comprised of a suite of protocols [7, 9, 10] developed to ensure the security services of integrity, confidentiality, and authentication of data communications over an IP network [1, 3]. While the flexibility of the IPsec standards has drawn the interest of the commercial sector, this same flexibility has resulted in several problems being identified with the protocols as a result of their complexity [8]. As with other security systems, poor maintenance can easily lead to a critical system failure [5].

IPsec may be used in three different security domains – Virtual Private Networks (VPNs), application level security, and routing security. At this time, IPsec is predominately used in VPNs. When used in application level security or routing security, IPsec is not a complete solution and must be coupled with other security measures in order to be effective, hindering its deployment in these domains [4].

2 IPsec Operation

IPsec has two modes of operation – *Transport Mode* and *Tunnel Mode*. When operating in *Transport Mode*, the source and destination hosts must directly perform all cryptographic operations. Encrypted data is sent through a single tunnel that is created with Layer 2 Tunneling Protocol (L2TP). Data (ciphertext) is created by the source host and retrieved by the destination host. This mode of operation establishes end-to-end security. When operating in *Tunnel Mode*, special gateways also perform cryptographic processing in addition to the source and destination hosts. Here, many tunnels are created in series between gateways, establishing gateway-to-gateway security [6]. When using either of these modes, it is important to provide all gateways with the ability to verify that a packet is real and to authenticate the packet at both ends. Any invalid packets must be dropped [2].

Two types of Data Packet Encodings (DPEs) are required in IPsec. These are the Authentication Header (AH) and the Encapsulating Security Payload (ESP) DPEs. These encodings offer network level security for the data [4]. The AH provides authenticity and integrity of the packet. The authentication is made available through keyed hash functions, also known as MACs. This header also prohibits illegal modification and has the option of providing anti-replay security. The AH can establish security between multiple hosts, multiple gateways, or multiple hosts and gateways, all implementing AH

[7]. The ESP header provides encryption, data encapsulation, and data confidentiality. Data confidentiality is made available through symmetric key encryption [3].

During its journey through the various tunnels and gateways, additional headers are added to the packet. On each pass through a gateway, a datagram is wrapped in a new header. Included in this header is the Security Protection Index (SPI). The SPI specifies the algorithms and keys that were used by the last system to view the packet. The payload is also protected in this system because any change or error in the data will be detected and will cause the receiving party to drop the packet. The headers are applied at the beginning of each tunnel and then they are verified and removed at the end of each tunnel. This method prevents the build-up of unnecessary overhead [1].

An important part of IPsec is the Security Association (SA). The SA uses the SPI number that is carried in the AH and ESP to indicate which SA was used for the packet. An IP Destination Address is also included to indicate the endpoint – this may be a firewall, router, or end user. An SA Database (SAdb) is used to store all SAs that are used. A Security Policy (SP) is used by the SAdb to indicate what the router should do with the packet. Three examples include dropping the packet altogether, dropping only the SA, or substituting a different SA. All of the SPs in use are stored in an SP Database (SPdb) [2].

3 IPsec Implementation

Most IPsec implementations are composed of three main parts – the SA Database (SAdb) and its management routines, the IPsec protocol engine, and the cryptographic transforms and algorithms [12]. The SAdb management routines are used to enforce specific system policies, including the type of information that the system is willing to send or receive in any direction. These policies typically vary from system to system. An example of such a system of policies may be found in the NIST Cerberus project:

> "By default, the system will always allow IPsec protected packets in and out of the system (as long as they have a relevant SA). In addition, through the system policy mechanism, allow non-protected packets to be sent or received. In this mode, if an SA is specified for a particular host, IPsec protected packets will be sent and in-bound IPsec protected packets will be decapsulated. At the same time, non-protected packets will be sent and received if no SA is specified. This mode is good for testing SAs and learning how to use the software and how is fits within you network architecture. Unfortunately, it does little to protect your system. For better control an additional policy mode allowing the user to specify NULL SAs is provided. A NULL SA is a place holder within the SADB that allows non-protected packets to be sent and received. This allows a user to poke small holes that will allow the system to communicate with other non-IPsec systems (e.g. DNS, NFS, etc). When using this mode (without the non-protected mode) the system will only communicate to those systems with SAs or NULL SAs [12]."

The IPsec protocol engine usually consists of two separate elements – one for outgoing data and one for incoming data. These elements control the application of the input and output cryptographic algorithms to the data, perform verification of authentication data, and modify the IP headers if necessary [12].

The cryptographic algorithms are very closely tied to the cryptographic transforms and are usually developed together. These are strictly math-oriented functions and are usually developed by a third party, separate from the IPsec implementation.

The software implementation of IPsec occurs predominately at the Operating System (OS) level. An OS level implementation has many advantages. First, this type of implementation enables developers to write applications that take advantage of IPsec. Because developers do not have to develop their own IPsec module for their applications, there is a reduced likelihood of implementation errors. Security issues also play a significant role in choosing to implement IPsec at the OS level. If applications are required to access IPsec through the OS, it is less likely that malicious applications will be able to take advantage of any unforeseen ambiguities in the IPsec protocol. If problems are found with the IPsec protocol, one update to the OS is all that would be required versus multiple patches to individual applications.

However, there are a number of arguments for implementing all or part(s) of the IPsec protocol at the application level (or in hardware) when developing custom applications. Examples include software applications that target an OS that does not incorporate IPsec or embedded systems that must incorporate IPsec while running either an OS that does not include IPsec or no OS at all. Custom applications such as these are numerous and present a significant implementation problem when considering the need to maintain and upgrade existing and future products.

4 IPv4 Networks Versus IPv6 Networks

The continuing upgrade from IPv4 networks to IPv6 networks has affected many protocols, including IPsec. One of the main reasons contributing to the need for IPv6 networks is the limited 32-bit address field available in IPv4 networks. Further exacerbating this limitation is the fact that the address field is not used to its full efficiency. IPv6 networks provide a 128-bit address space and manage these addresses more efficiently than IPv4 networks. As an example, IPv4 networks must manually renumber to connect to the Internet while IPv6 networks perform this function automatically [13]. Another issue with IPv4 networks is the need for Network Address Translation (NAT) for internal devices on a network (for example, a home network) [14].

Many new applications are being developed that will require IPv6 networks because of the required address space. Common home appliances are already being manufactured that will require IP addresses. Some of these appliances include the “Internet Fridge” by LG Electronics and the Sharp RE-M210 Internet-capable microwave [14]. It is expected that within five years, automobile telemaintenance and telediagnostic applications will require one IP address for every new car manufactured [15]. These products demonstrate

an increasing demand for an expanded address space that is not possible when using IPv4 networks.

Finally, it is critical to note that IPv6 networks specify mandatory support of IPsec. This enables IPv6 networks to provide all of the features that come with IPsec, including strong integrity and authentication of IP packets through the AH header and strong integrity and confidentiality of IP packets through the ESP header [13].

5 Problems with IPsec

In some cases, direct end-to-end communication, i.e. *Transport Mode*, is not possible. The following example demonstrates a situation in which this is the case:

> "In a large distributed system or inter-domain environment, the diversified regional security policy enforcement can create significant problems for end-to-end communication. In the above example, suppose FW1 needs to examine traffic content for the purpose of intrusion detection and a policy is set up at FW1 to deny all encrypted traffic to enforce its content examination requirement. Yet, H1 and H2 build a direct tunnel without awareness of existence of the firewall and its policy rules. Therefore, all the traffic will be dropped by FW1. The scenario shows that each policy satisfies its corresponding requirement while all policies together can cause conflicts [5]."

One of the biggest drawbacks of IPsec is its complexity. While IPsec's flexibility has contributed to its popularity, it also leads to confusion and has led to security experts to state that "IPsec contains too many options and too much flexibility [8]." Much of IPsec's flexibility and complexity may be attributed to the fact that IPsec was developed via a committee process. Due to the political nature of committees, additional features, options, and flexibility are often added to standards to satisfy various factions of the standardization body [8]. This process stands in stark contrast to the standardization process used in the development of the Advanced Encryption Standard (AES), the replacement for the Data Encryption Standard (DES), which expired in 1998 [11]:

> "It is instructive to compare this to the approach taken by NIST for the development of AES. Instead of a committee, NIST organized a contest. Several small groups each created their own proposal, and the process is limited to picking one of them. At the time of writing there has been one stage of elimination, and any one of the five remaining candidates will make a much better standard than any committee could ever have made [8]."

Moreover, much of the documentation for IPsec is complex and confusing. No overview or introduction is provided and nowhere are the goals of IPsec identified. The user must assemble the pieces and try to make sense of documentation that may be described as difficult to read, at best. To illustrate the frustration a user must endure, consider the ISAKMP specifications. These specifications are missing many key explanations, contain numerous errors, and contradict themselves in various locations [1, 8].

However, while IPsec may not be perfect, it is considered to be a significant improvement versus previously available security protocol. As an example, consider the popular security system Secure Sockets Layer (SSL). While SSL is widely deployed in various applications, SSL is inherently limited in that it is used on the transport/application layer, requiring modifications to any application that wants to include the ability to use SSL. Because IPsec is used in layer 3, it only requires modification to the OS rather than modifications to the applications that employ IPsec [3].

5 Conclusions

A general overview of the operation of IPsec has been presented and the advantages and disadvantages of the architecture have been discussed. IPsec incorporates all of the most commonly employed security services, including authentication, integrity, confidentiality, encryption, and non-repudiation. However, the major drawbacks to IPsec are its complexity and the confusing nature of its associated documentation. In spite of these various drawbacks, IPsec is believed by many to be one of the best security systems available. It is hoped that considerable improvement will be evidenced in future revisions of IPsec and that the problems identified with the architecture will be remedied.

References

[1] J. D. Guttman, A. L. Herzog, and F. J. Thayer, "Authentication and Confidentiality via IPsec," Proceedings of the 6th European Symposium on Research in Computer Security – ESORICS 2000, LNCS 1895, available at http://www.ccs.neu.edu/home/guttman/esorics-ipsec.pdf, June 30 2000.

[2] C.-L. Wu, S. F. Wu, and R. Narayan, "IPSec/PHIL (Packet Header Information List): Design, Implementation, and Evaluation," Proceedings of the Tenth International Conference on Computer Communications and Networks, available at http://www.cs.ucdavis.edu/~wu/publications/314-PHIL.pdf, October 15-17 2001.

[3] R. Perlman and C. Kaufman, "Analysis of the IPSec Key Exchange Standard," Proceedings of the Tenth IEEE International Workshop on Enabling Technologies: Infrastructure for Collaborative Enterprises – WET ICE 2001, available at http://sec.femto.org/wetice-2001/papers/radia-paper.pdf, 2001.

[4] IP Security Paper Summary, available at http://staging.denison.edu/~bressoud/cs402-f03/summary-question/ipsec-summary.pdf.

[5] Z. Fu, S. F. Wu, H. Huang, K. Loh, F. Gong, I. Baldine, and C. Xu, "IPSec/VPN Security Policy: Correctness, Conflict Detection, and Resolution," International Workshop on Policies for Distributed Systems and Networks – POLICY 2001,

LNCS 1995, available at http://www.cs.ucdavis.edu/~wu/publications/ipsecpolicy.PDF, 2001.

[6] D. Shinder, "Securing Data in Transit with IPsec," available at http://www.windowsecurity.com/articles/Securing_Data_in_Transit_with_IPSec.html, Feb 17 2003.

[7] S. Kent and R. Atkinson, *Security Architecture for the Internet Protocol*, IETF Network Working Group RFC 2401, available at http://www.ietf.org/rfc/rfc1825.txt, November 1998.

[8] N. Ferguson and B. Schneier, "A Cryptographic Evaluation of IPsec," Counterpane Internet Security, Inc., available at: http://www.counterpane.com/ipsec.html, 1999.

[9] S. Kent and R. Atkinson, *IP Authentication Header*, IETF Network Working Group RFC 2402, November 1998.

[10] S. Kent and R. Atkinson, *IP Encapsulating Security Payload*, IETF Network Working Group RFC 2406, November 1998.

[11] B. Schneier, *Applied Cryptography*, John Wiley & Sons Inc., New York, New York, USA, 2nd edition, 1996.

[12] NIST Cerberus, An IPsec Reference Implementation for Linux, available at http://snad.ncsl.nist.gov/cerberus/

[13] Internet Protocol Version 6 versus IPv4, available at http://ipsit.bu.edu/sc546/sc441Spring2003/ipv6/v4_v6.htm

[14] T. Chown, "IPv6 in the Home Makes Sense", available at http://www.ec.ipv6tf.org/PublicDocuments/020911-ipv6-home-06.pdf

[15] J. Prevost, "IPv6 today: Why, How?", available at http://www.renater.fr/Video/2002ATHENS/P/JP-IPv6/JP-IPv6.pdf