A Multi-Layer IPsec Protocol

Yongguang Zhang Bikramjit Singh
HRL Laboratories, LLC
{ygz,bsingh}@hrl.com

Abstract

IPsec [KA98c] is a suite of standard protocols that provides security services for Internet communications. It protects the entire IP datagram in an "end-to-end" fashion; no intermediate network node in the public Internet can access or modify any information above the IP layer in an IPsec-protected packet. However, recent advances in internet technology introduce a rich new set of services and applications, like traffic engineering, TCP performance enhancements, or transparent proxying and caching, all of which require intermediate network nodes to access a certain part of an IP datagram, usually the upper layer protocol information, to perform flow classification, constraint-based routing, or other customized processing. This is in direct conflict with the IPsec mechanisms. In this research, we propose a multi-layer security protection scheme for IPsec, which uses a finer-grain access control to allow trusted intermediate routers to read and write selected portions of IP datagrams (usually the headers) in a secure and controlled manner.

1 Introduction

The Internet community has developed a mechanism called *IPsec* for providing secure communications over the public Internet. IPsec can provide data integrity, origin authentication, data confidentiality, access control, partial sequence integrity, and limited traffic flow confidentiality services for communications between any two networks or hosts [KA98c]. By addressing the security issues at the IP layer and rendering the security services in a transparent manner, IPsec attempts to relieve software developers from the need to implement security mechanisms at different layers or for different Internet applications. Arguably, IPsec is the best available mechanism for Virtual Private Networks (VPN) and secure remote accesses.

1.1 The Protection Model in IPsec

The fundamental concept behind the IPsec technology is as follows. The path between an IP datagram's source and destination is divided into three segments (see Figure 1) — the protected and trustworthy local network at the source (e.g., a company's private LAN), the untrustworthy public Internet segment, and the protected and trustworthy local network at the destination. The IPsec architecture places a security gateway (here G_1 and G_2) at each boundary between a trustworthy and an untrustworthy network. Initially, G_1 at the source establishes a security association with G_2 on the destination side, which is a security relationship that involves negotiation of security services and shared secrets. Before an IP datagram (from S to D) is sent to the untrustworthy Internet, the security gateway (G_1) encrypts and/or signs the datagram using an IPsec protocol. When it reaches the security gateway at the destination side (G_2), the datagram is decrypted and/or checked for authentication, before it is forwarded to the destination (D). In some cases, the trustworthy local network on either side can be omitted, and the source or destination host can perform encryption, authentication and other security-gateway functions itself.

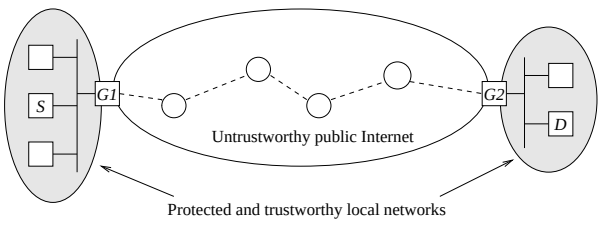

Figure 1: System Model

The IPsec architecture uses two protocols to provide traffic security – AH (Authentication Header) [KA98a] and ESP (Encapsulating Security Payload) [KA98b]. AH provides integrity and authentication without confidentiality; ESP provides

confidentiality, with optional integrity and authentication. Each protocol supports two modes of use: *transport mode* and *tunnel mode*. Transport mode provides protection primarily for upper layer protocols, while in tunnel mode the protection applies to the entire IP datagram.

The granularity of security protection in the IPsec architecture is at the datagram level. It treats everything in an IP datagram after the IP header as one integral unit. Usually, an IP datagram has three consecutive parts – the IP header (for routing purposes only), the upper layer protocol headers (for example, the TCP header), and the user data (for example, the TCP data). In transport mode, an IPsec protocol header (AH or ESP) is inserted in after the IP header and before the upper layer protocol header to protect the upper layer protocols and user data. In tunnel mode, the entire IP datagram is encapsulated in a new IPsec packet (a new IP header followed by an AH or ESP header). In either case, the upper layer protocol headers and data in an IP datagram are protected as one indivisible unit (see Figure 2).

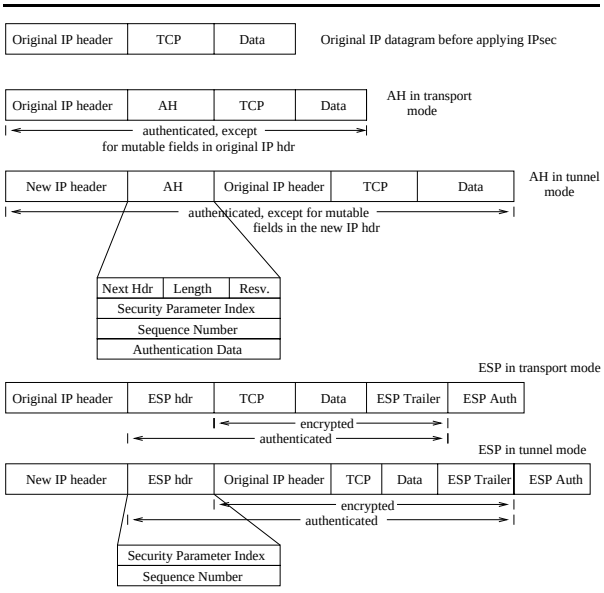

Figure 2: The Protocol Formats of IPsec-protected IPv4 Packets (assuming TCP)

The keys used in encryption and authentication are shared only by the sender-side and receiver-side security gateways. All other nodes in the public Internet, whether they are legitimate routers or malicious eavesdroppers, see only the IP header and will not be able to decrypt the content, nor can they tamper with it without being detected. Traditionally, the intermediate routers do only one thing – forward packets based on the IP header (mainly the destination address field); IPsec's "end-to-end" model is well-suited to this layering paradigm.

1.2 Limitations of End-to-End Security

However, this protection model and its strict layering principle are unsuitable for an emerging class of new networking services and applications for the next generation Internet. Unlike in the traditional minimalist Internet, intermediate routers begin to play more and more active roles. They often rely on some information about the IP datagram payload, such as certain upper layer protocol header fields, to make sophisticated routing decisions. In other words, routers can now participate in a layer above the IP. Examples of such active networking techniques are:

- Internet traffic engineering. The Internet is moving towards active traffic engineering to meet increasing demand for bandwidth and rich services. Routers/switches will support per-flow and class-based queueing to give fair bandwidth access to all users. A QoS guarantee will be provided to traffic flows generated by paying customers. Router-based congestion control mechanisms, such as Random Early Detection (RED) [FJ93] with penalty box [FF99], also require intermediate nodes to discriminate between traffic flows. Depending on the granularity used in defining a "flow," certain nodes in the middle of the network may need access to information in the upper layer protocols, such as TCP/UDP port numbers, to classify packets into flows before applying discriminating operations.

- Transport-aware link layer mechanisms. The global Internet has accommodated a very wide range of link technologies, but certain transport protocols like TCP have not achieved optimal performance when operated over a path that includes lossy wireless links or long-delay satellite links. For example, in a recent paper [BPSK97], Balakrishnan proves that, to significantly improve the TCP's performance over a wireless link, the base station at the lossy link must be aware of the TCP state information in each passing flow, and deliberately delay or drop certain types of TCP packets. Such link-layer

mechanisms for TCP performance improvement (often referred to as *TCP Performance Enhancing Proxies* or TCPPEP [BKGM00]) require intermediate nodes to access and sometimes modify the upper layer protocol headers.

- Application-layer proxies/agents. Some Internet routers can provide application-layer services for performance gains. For example, an intermediate router can become a transparent web proxy when it snoops through the TCP and HTTP header of a bypassing IP datagram to determine the URL request, and serves it with the web page from the local cache. It is transparent to end-users but boosts the responsiveness because the delivery paths for web requests and data between the intermediate router and the web site server are eliminated.

- Active networks. Going one step further, the active network architecture is a new networking paradigm in which the routers perform customized computation on the data flowing through them. A number of experimental active network systems have been developed and they can be run over the Internet. In this architecture, a single IP datagram carries not only upper-layer protocol headers and user data, but also a "method" – a set of executable instructions to be interpreted by the intermediate routers, for describing, provisioning, or tailoring network resources and services in order to achieve the delivery and management requirements. Obviously then, the "method" portion of the IP datagram ought not to be encrypted "end-to-end."

- Traffic Analysis. Many network operators actively monitor the traffic for accounting or for intrusion detection purposes. Usually, such monitoring requires logging of certain upper layer protocol information, like TCP/UDP ports. Many firewalls that protect local networks also depend on such information to deny unauthorized traffic.

All these mechanisms require intermediate network nodes to access information encoded in the IP datagram payload, but the current IPsec technology advocates end-to-end security and prevents such access. This fundamental conflict [NBB99] makes it a very difficult problem to provide both security and extensibility in one unified platform.

1.3 Problem Statement

The goal of this research is to develop a security scheme that supports the above new network services and applications under the IPsec framework. The new scheme should *grant trusted intermediate routers a secure, controlled, and limited access to a selected portion of certain IP datagram, while preserving the end-to-end security protection to user data.*

2 Approaches

We have investigated three ways to solve the problem – replacing IPsec with a transport-layer security mechanism, using a transport-friendly ESP format, and developing a multi-layer protection model for IPsec.

The first approach, replacing IPsec with a *transport-layer mechanism*, circumvents the problem of intermediate nodes not being able to access the encrypted TCP headers, yet introduces certain other difficulties. There are actually several transport-layer security mechanisms available today, including SSL (most notably used in Netscape and other WWW applications) or TLS (a proposed IETF standard [DA99]). Both SSL and TLS encrypt the TCP data while leaving the TCP header in unencrypted and unauthenticated form so that intermediate nodes can make use of the TCP state information encoded in the TCP header. However, letting the entire TCP header appear in clear text exposes several vulnerabilities of the TCP session to a variety of TCP protocol attacks (in particular traffic analysis), because the identity of sender and receiver are now visible without confidentiality protection.

Alternatively, it is possible to tunnel one security protocol within another, such as SSL/TLS inside an IPsec ESP – letting SSL/TLS protect the TCP data and ESP protect the TCP header. However, there is a problem here too because ESP encrypts both TCP header and TCP payload (SSL/TLS-protected data) as a whole. Thus, the encryption/authentication/decryption has to be done twice on the TCP data part, an unnecessary waste of resources. The intermediate router, for example, must decrypt the entire packet to access just the TCP header information.

The second approach is to develop a *transport-friendly ESP* (TF-ESP) protocol format for IPsec. Proposed by Steve Bellovin of AT&T Labs [Bel99], TF-ESP modifies the original ESP protocol to include limited TCP state information, such as flow identifications and sequence numbers, in a disclosure header outside the encryption scope (but authenticated). This approach will work well for some TCP PEP mechanisms such as TCPPEP for wireless network (e.g., TCP snooping), but it may not suite other mechanisms that need a write access, such as TCPPEP for satellite networks [ZDRD97, BKGM00]. To support TCPPEP for satellite networks, the TCP state information also needs to be placed outside the authentication scope. Without proper integrity protection, this can be dangerous. Further, the unencrypted TCP state information is made available universally, including to untrustworthy nodes, which creates vulnerability for possible attacks. In addition, TF-ESP is not flexible enough to support all upper-layer protocols.

Since the above two approaches both have limitations, we thus propose a third approach – to develop a *multi-layer security protection scheme* for IPsec. The idea is to divide the IP datagram into several parts and apply different forms of protection to different parts. For example, the TCP payload part can be protected between two end points while the TCP/IP header part can be protected but accessible to two end points plus certain routers in the network. The rest of this paper will describe the principle, the design and an implementation of this approach.

3 The Principle of Multi-Layer Security Protection

Our approach is called ML-IPsec (Multi-Layer IPsec). It uses a multi-layer protection model to replace the single end-to-end model. Unlike IPsec where the scope of encryption and authentication apply to the entire IP datagram payload (sometimes IP header as well), our scheme divides the IP datagram into zones. It applies different protection schemes to different zones. Each zone has its own sets of security associations, its own set of private keys (secrets) that are not shared with other zones, and its own sets of access control rules (defining which nodes in the network have access to the zone).

When ML-IPsec protects a traffic stream from its source to its destination, the first IPsec gateway (or source) will re-arrange the IP datagram into zones and apply cryptographic protections. When the ML-IPsec protected datagram flows through an authorized intermediate gateway, a certain part of the datagram may be decrypted and/or modified and re-encrypted, but the other parts will not be compromised. When the packet reaches the last IPsec gateway (or destination), ML-IPsec will be able to reconstruct the original datagram. ML-IPsec defines a complex security relationship that involves both the sender and the receiver of a security service, but also selected intermediate nodes along the traffic stream.

For example, a TCP flow that desires link-layer support from the network can divide the IP datagram payload into two zones: TCP header and TCP data. The TCP data part can use an end-to-end protection with keys shared only between the source and the destination (hosts or security gateways). The TCP header part can use a separate protection scheme with keys shared among the source, the destination, and certain trusted intermediate node. (See Figure 3.) This way, no one in the public Internet other than the source, the destination and the trusted intermediate nodes has access to TCP header or TCP data, and no one other than source and destination (not even the trusted intermediate node) has access to TCP data.

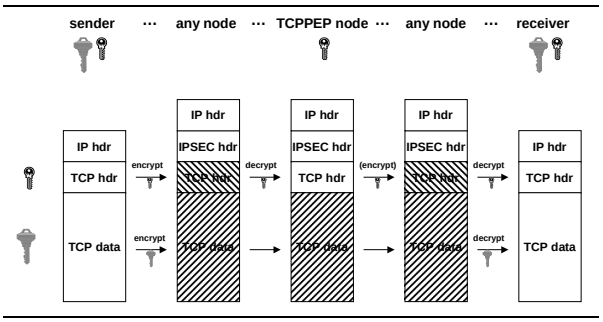

Figure 3: Multi-Layer Protection Model for TCP

This scheme in effect provides a finer-grain access control to the IP datagram. Since ML-IPsec allows network operators and service providers to grant intermediate nodes limited access to IP datagram contents parts (such as TCP header), such access must be granted in a secure and controllable way. The identity of the intermediate nodes must be authenticated (using an out-of-band mechanism such as a public-key infrastructure) to prevent any man-in-the-middle attack. After authentication, keys or

shared secrets corresponding to the authorized IP datagram zones must be distributed to the intermediate nodes, also using out-of-band mechanisms like IKE [HC98].

4 ML-IPsec Design Details

The architecture of ML-IPsec embraces the notion of zones, a new type of security association, the new AH and ESP header formats, and the inbound/outbound processing of ML-IPsec protocol packets. It is designed to be fully compatible with the original IPsec in both protocol formats and processing software.

4.1 Zones

A zone is any portion of IP datagram under the same security protection scheme. The granularity of a zone is 1 octet. The entire IP datagram is covered by zones, except for the IP header in the transport modes, but zones cannot overlap. Using the same TCP example, the portion of the IP datagram that contains TCP header (21st to 40th octet) is Zone 1, and the TCP data portion (41st and above octet) is Zone 2 (assuming transport mode and no TCP options).

A zone need not be a continuous block in an IP datagram, but each continuous block is called a *subzone*. A *zone map* is a mapping relationship from octets of the IP datagram to the associated zones for each octet. Figure 4 below shows a sample zone map.

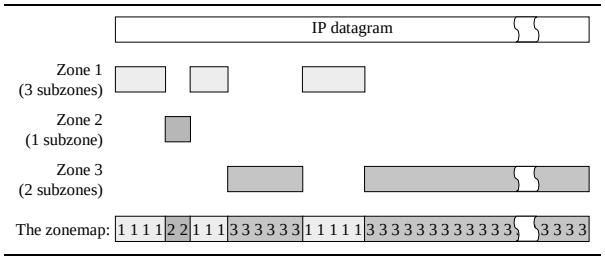

Figure 4: A Sample Zone Map

The zone map is a constant in a security relationship. That is, the zone boundaries in each IP datagram must remain fixed in the lifetime of the security association; otherwise, it will be extremely difficult to do zone-by-zone decryption and authentication. Since IP datagrams are variable in length, the zone that covers the last part of the datagram, usually the user data, should also be variable in size. Zone 3 of the above is an example. It is also possible, theoretically, to define a phantom zone that does not correspond to any byte in an IP datagram.

4.2 Security Association (SA)

4.2.1 Original SA for IPsec

Security Association (SA) is a key concept in the IPsec technology [KA98c]. It is a one-way relationship between a sender and a receiver that affords security services. Each SA defines a set of parameters including the sequence number and anti-replay window for anti-replay service, the protocol mode (transport or tunnel), the lifetime of the SA, the path MTU and other implementation details. For authentication services in AH or ESP, each SA also defines the choice of cryptographic algorithm, the crypto-keys, key lifetimes and related parameters. For encryption services in ESP, each SA further defines the choice of encryption algorithm, the encryption keys, the initial values, key lifetimes, etc. When an outbound IP datagram passes the security gateway, the IPsec module first compares the values of the appropriate fields in the IP datagram (the selector fields) against a set of predefined policies, called SA selectors, in the Security Policy Database (SPD). It then determines the SA for this datagram if any, and does the required security processing (e.g., encryption). When an inbound IPsec datagram passes the security gateway, the IPsec module uses the SPI (Security Parameter Index) field to determine the SA for this datagram and performs security processing (e.g., decryption). Figure 5 gives a simple illustration of how these pieces are connected together in the IPsec architecture.

4.2.2 Composite SA for ML-IPsec

SA in the original IPsec defines a simple security relationship from the sender to the receiver that affords the protection service. ML-IPsec however requires a much more complex security relationship to include sender and receiver, as well as the selected intermediate nodes. Since the security service is zone-by-zone, conceptually we can use an indi-

DOCKET ALARM

Explore Litigation Insights

Docket Alarm provides insights to develop a more informed litigation strategy and the peace of mind of knowing you're on top of things.

Real-Time Litigation Alerts

Keep your litigation team up-to-date with **real-time alerts** and advanced team management tools built for the enterprise, all while greatly reducing PACER spend.

Our comprehensive service means we can handle Federal, State, and Administrative courts across the country.

Advanced Docket Research

With over 230 million records, Docket Alarm's cloud-native docket research platform finds what other services can't. Coverage includes Federal, State, plus PTAB, TTAB, ITC and NLRB decisions, all in one place.

Identify arguments that have been successful in the past with full text, pinpoint searching. Link to case law cited within any court document via Fastcase.

Analytics At Your Fingertips

Learn what happened the last time a particular judge, opposing counsel or company faced cases similar to yours.

Advanced out-of-the-box PTAB and TTAB analytics are always at your fingertips.

API

Docket Alarm offers a powerful API (application programming interface) to developers that want to integrate case filings into their apps.

LAW FIRMS
Build custom dashboards for your attorneys and clients with live data direct from the court.

Automate many repetitive legal tasks like conflict checks, document management, and marketing.

FINANCIAL INSTITUTIONS
Litigation and bankruptcy checks for companies and debtors.

E-DISCOVERY AND LEGAL VENDORS
Sync your system to PACER to automate legal marketing.