Network Working Group
INTERNET DRAFT

Y. Zhang
HRL Labs

October 1999

Expires April 2000

Multi-Layer Protection Scheme for IPSEC
<draft-zhang-ipsec-mlipsec-00.txt>

Status of this Memo

Abstract

This document describes a multi-layer protection scheme for IPSEC that applies separate encryption/authentication with different keys on different parts of an IP datagram. It allows certain intermediate routers to have limited and controllable access to part of IP datagram (usually headers) but not the user data.

1. Introduction

Recent advances in internetwork technology introduce a rich set of new services and applications, like router-based congestion controls, or TCP performance enhancements, or transparent proxies, all of which require intermediate network nodes to access certain part of an IP datagram, usually the upper layer protocol information, to perform intelligent routing or customized processing. Many of these mechanisms have already found widely used, such as per-flow queueing,

multi-field diffserv, TCPPEP for wireless/satellite networks, and NAT for ISPs. Other promising mechanisms include layer-4/5/6/7 routing/switching, or even ``active networks.'' Various IETF working groups or BOFs are addressing such needs, like PEP, PILC, TF-ESP, etc. Unfortunate, all these are in direct conflict with the IETF standard mechanism for providing security services -- IPSEC. IPSEC [RFC 2401] protects the entire IP datagram in an ``end-to-end'' fashion; no intermediate node in the public Internet can access or modify any information above the IP layer in an IPSEC-protected packet, which in effect disable all the above new services and applications. This memo specifies a multi-layer security protection scheme for IPSEC, which uses a finer-grain access control to allow trusted intermediate routers read and write a selected portion of IP datagram, in a secure and controlled manner. The goal is to support the above new services and applications, and preserves the end-to-end security protection for user data.

1.1 Background of IPSEC operation

IPSEC uses two protocols to provide traffic security -- AH (Authentication Header) [RFC 2402] and ESP (Encapsulating Security Payload) [RFC 2406]. 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 IPSEC is at the datagram level. IPSEC treats everything in an IP datagram after the IP header as one integrity unit. Usually, an IP datagram has three consecutive parts -- the IP header (for routing purpose only), and the upper layer protocol headers (e.g., the TCP header), and the user data (e.g., TCP data). In transport mode, an IPSEC 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). Either case, the upper layer protocol headers and data in an IP datagram are protected as one indivisible unit.

The access control of IPSEC is through the distribution of keys used in authentication and encryption. Whoever has the keys have read

and/or write access to the entire IP datagram. Normally, the keys 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 it without

being detected. This end-to-end model works well if the internet routers do only one thing - forwarding packets based on IP header (mainly the destination address field), but not if intermediate routers perform extra customized processing or intelligent routing based on some content of the datagram, such as the upper-layer protocol headers. An example of such cases is multi-field diffserv, where intermediate routers uses the TCP header information to classify flows and to expedite packet forwarding.

Security Association (SA) [RFC 2401] is a key concept in IPSEC. 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 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, IPSEC 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 IPSEC processing (i.e., encryption). When an inbound IPSEC datagram passes the security gateway, IPSEC uses the SPI (Security Parameter Index) field to determine the SA for this datagram and performs IPSEC processing (i.e. decryption).

2. A New Multi-Layer Security Protection Model

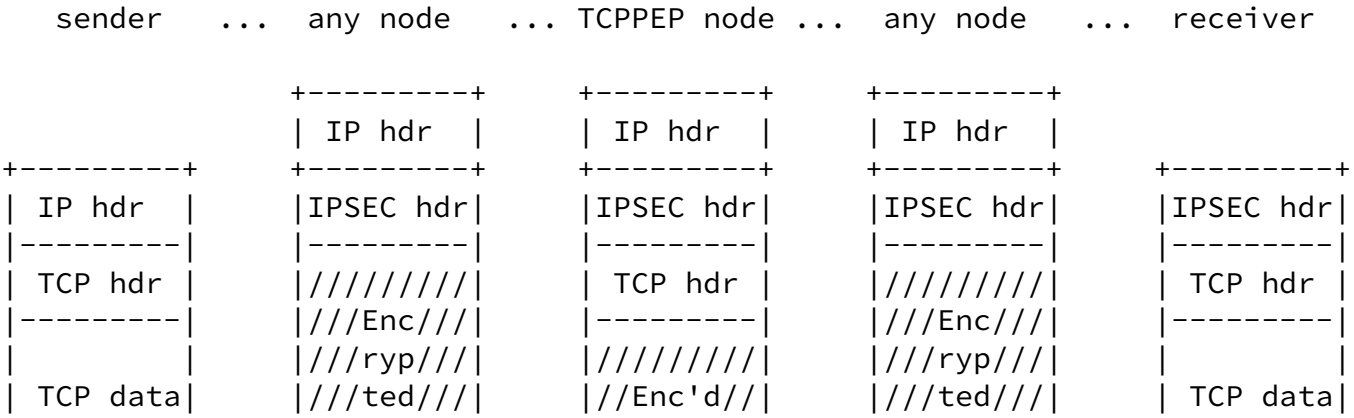
ML-IPSEC is an extension to IPSEC that adds a finer grain access control. It uses a multi-layer security protection model to replace the single end-to-end model. Under IPSEC, the scope of encryption and authentication apply to the entire IP datagram payload (sometimes IP header as well). In ML-IPSEC, an IP datagram is divided into more

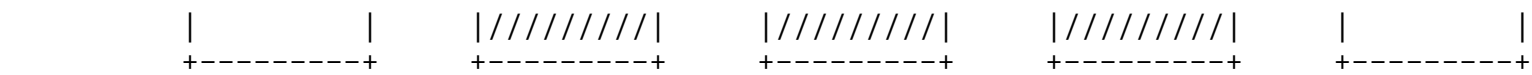
than one ``zones.'' Different security relationship are defined for different zones. Each zone has its own sets of security associations, its own set of private keys (secrets) that are not shared with other zones, and its own sets access control rules (defining which nodes have access to the zone).

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 applies cryptographic protections. When the ML-IPSEC protected datagram flow through an authorized intermediate gateway, certain part of the datagram may be decrypted and/or modified and re-encrypted, but the other part 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 an security service, but also selected intermediate nodes along the traffic stream.

For example, in a TCPPEP (TCP Performance Enhancement Proxy) application, the IP datagram payload is divided into two zones: TCP header and TCP data. The TCP data part uses an end-to-end protection with keys shared only between source and destination (either end hosts or IPSEC gateways). The TCP header part uses a separate protection scheme with keys shared among source, the destination, and certain trusted intermediate node (that performs TCPPEP). This way, no one in the public Internet other than source, destination or the trusted intermediate node 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.





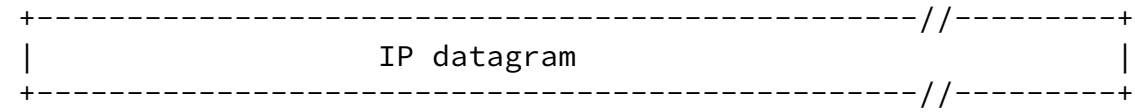
Since ML-IPSEC allows network operators and service providers to grant limited access of IP datagram contents (such as TCP header) to intermediate nodes, such access must be granted in a secure and controllable way. The identity of the intermediate nodes must be authenticated (using an out-of-band mechanism such as public-key infrastructure) to prevent any man-in-the-middle attack. This memo does not specify the authentication procedure. It however recommends the same authentication mechanisms used for the original IPSEC.

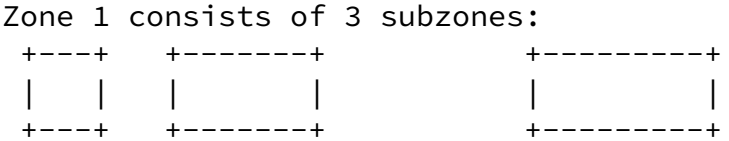
3. Zones

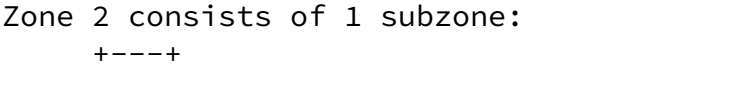
A zone is the portion of IP datagram under the same security protection scheme. The granulative of a zone is 1 octet. The entire IP datagram is covered by zones, except the IP header in IPSEC transport mode, but zones cannot overlap.

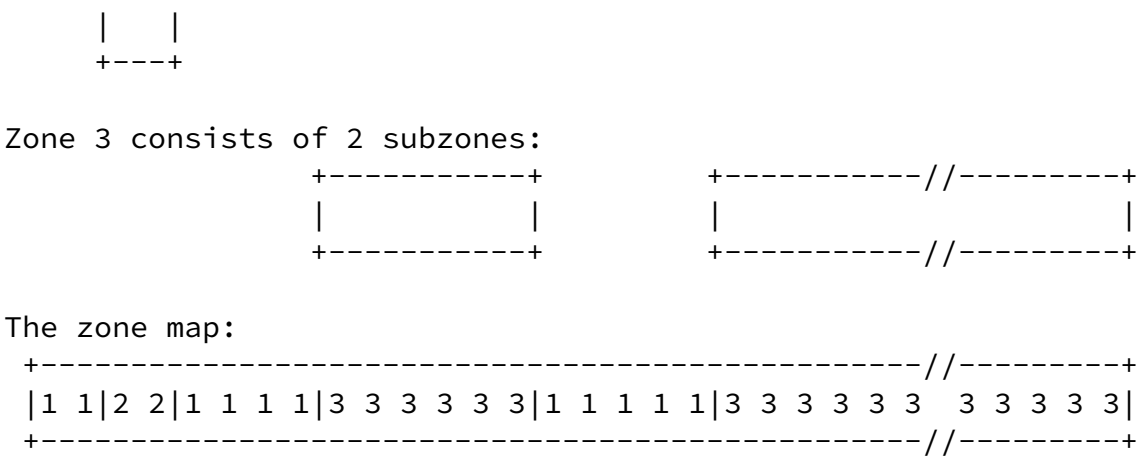
Using the multifield diffserv or TCPPEP as an example, the portion of IP datagram that contains TCP header (21st to 40th octet) is Zone 1, and the TCP data portion (41st and above octet) is Zone 2 (assuming transport mode and no TCP options).

A zone needs not to 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. The following figure illustrates an example zonemap.









The zone map is a constant in a security relationship. That is, the zone boundaries in each IP datagram must remain fixed in the life time 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. Composite SA

RFC 2401 defines a simple security relationship from the sender to the receiver that afford 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 individual security relationship to cover each zone, then build a composite relationship to cover the entire IP datagram. Mapping this idea to the basic Security Association (SA) concept, ML-IPSEC needs a new type of SA called ``Composite SA'' (CSA). CSA is a collection of SAs (per RFC 2401) that collectively afford a multi-layer security protection for the traffic stream.

A CSA has two elements. The first element is a zone map. The zone map specifies the coverage of each zone in an IP datagram. The zone map must be consistent in all nodes involved in the same ML-IPSEC

relationship.

The second element in a CSA is a zone list. A zone list is a list of SAs for all the zones. Each and every such SA is stored in the Security Association Database (SAD) [RFC 2401]. However, some of the fields are used different in ML-IPSEC than in RFC 2401. The following SAD fields, for example, are applicable only on the corresponding zone of the SA.

Lifetime of this Security Association.
AH Authentication algorithm, keys, etc.
ESP Encryption algorithm, keys, IV mode, IV, etc.
ESP Authentication algorithm, keys, etc.

The other SAD fields have no meanings in the zone level. With the exception of a designated SA in the zonelist, the following SAD fields are not use in other zonal SAs, although they may be initialized during the SA creation process.

Sequence Number Counter.
Sequence Counter Overflow.
Anti-Replay Window.
IPsec protocol mode.
Path MTU.

The designated SA however operates on these fields as defined in RFC 2401. The designated SA is a special SA in the zonelist, usually the first SA in the list. It is responsible for maintaining parameters for the IP datagram layer and ``represents'' the CSA in IPSEC processing.

The zone map and zone list can be stored with the designated SA as additional fields in the SAD, or, they can be store in a separate CSA

database. This is an implementation choice and it allows flexibility in adding ML-IPSEC features to an existing IPSEC implementation.

On inbound processing, if the traffic stream is under ML-IPSEC protection, the destination IP address, the IPsec protocol type, and the SPI identifies an entry in the SAD, which points to the designated SA of the CSA for this traffic stream. Or, under alternative implementation, the triplet identifies an entry in the

CSA database. By traversing CSA's zone list ML-IPSEC can further identifies the SA entries for all the zones.

On outbound processing, the Security Policy Database (SPD) [RFC 2401] will have a pointer to the designated SA or an entry in the CSA database. Same as in RFC 2401, the selectors will direct the outbound traffic to the proper SPD entry.

4.1 Access Control in a CSA

A CSA involves the sender, receiver, and all authorized intermediated nodes that collectively provide a multi-layer security protection for a traffic stream. Therefore, an instance of CSA must be created in each of these nodes before the ML-IPSEC service can commence. The zonemap must be distributed and remain the same for all nodes. Each CSA instance must have a designated SA, and the choice of designated SA must be consistent across all nodes.

However, the zone list need not be the same for all nodes. In principle, each zonal SA independently determines the access list for that zone; not all nodes will have access to all zones. If some node does not have access to a zone, the corresponding zonal SA in the zone list will be null. For a particular zonal SA, an instance must be created in each authorized node and stored in its SAD as a step in CSA creation. By determining which zonal SA to be created in which node, CSA enforces a multi-layer access control for an IP traffic stream.

No node is allowed to have null designated SA. That is, every nodes involved in an ML-IPSEC relationship must all have access to at least one zone, although, in principle, it is possible to include a phantom zone and define the designated SA on that zone. For convenient, we called the zone for which the designated SA is chosen the "designated zone".

4.2 An TCP Example

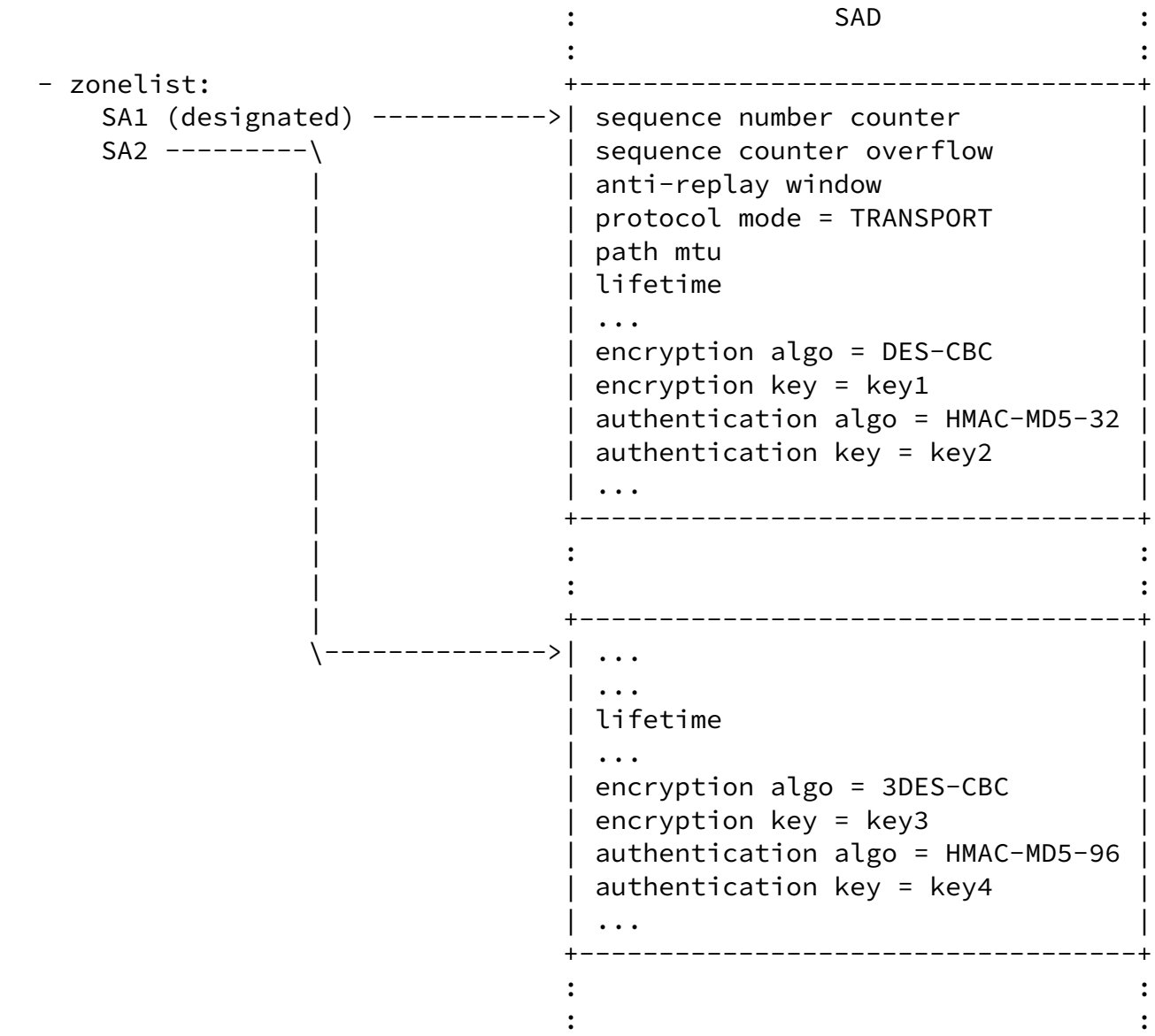
Here is an example to illustrate the concept of CSA. It is a traffic flow from Sender (the ultimate source or the outbound IPSEC gateway) to Receiver (the ultimate destination or the inbound IPSEC gateway),

passing through Gateway (an intermediate router providing diffserv or

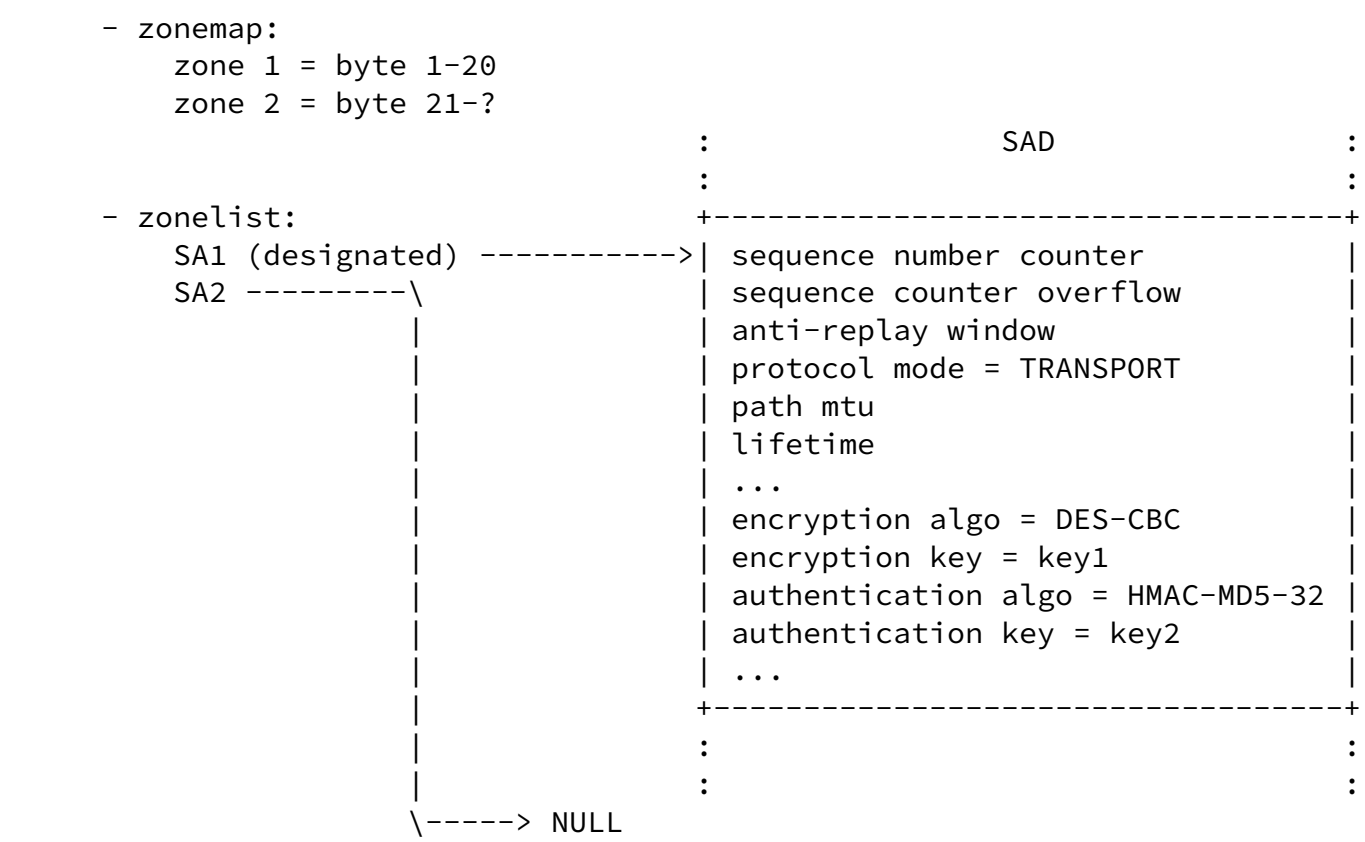
TCPPEP service). Let's assume the desired security service is ESP transport mode.

The corresponding CSA in Sender or Receiver will have the following elements:

- zonemap:
  zone 1 = byte 1-20
  zone 2 = byte 21-?
- zonelist:
  SA1 (designated)
  SA2



The corresponding CSA in Gateway will have the following elements:



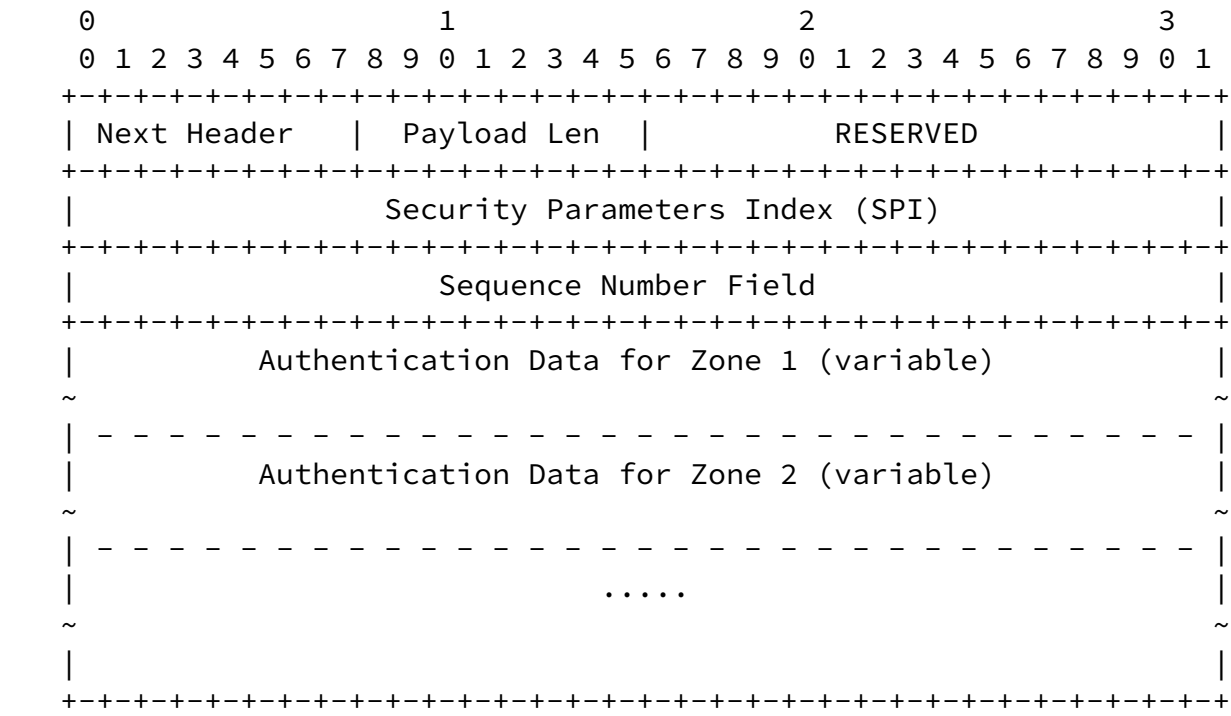
(HMAC-MD5-32 is a hypothetical keyed hash algorithm that produces a smaller 4-octet signature.)

5. AH and ESP Headers

The same security protocol formats, AH and ESP, are used in ML-IPSEC. Both AH and ESP have transport mode or tunnel mode, as indicated by the "protocol mode" field of the designated SA. Since the transport mode provides protection primary for

5.1 AH Headers

In ML-IPSEC, the protocol header format for AH is almost identical to IPSEC AH (RFC 2402), except that the Authentication Data section in AH can be further subdivided into zones:



The "Authentication Data" field is a variable-length field that contains several Integrity Check Values (ICVs) for this packet. The total length of this field is controlled by "Payload Len". The size of each ICV is determined by the authentication algorithm used in each zonal SA, but must be an integral multiple of 32 bits. The boundaries of these zonal authentication data can be derived from the CSA.

5.2 Integrity Check Value Calculation and Verification

The AH ICV calculation is rather different in ML-IPSEC. For the designated zone, the ICV is computed over:

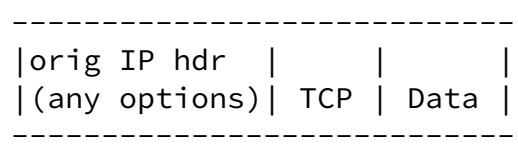
- IP header fields that are immutable in transit.

- the AH header (Next Header, Payload Len, Reserved, SPI, Sequence Number, and the Authentication Data (which is set to zero for this computation), and explicit padding bytes (if any))

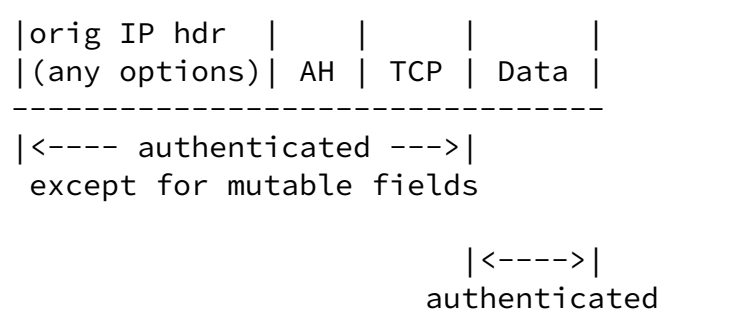
- all octets in the designated zone.

For other non-designated zones, the ICV is computed only over the octets of the zone.

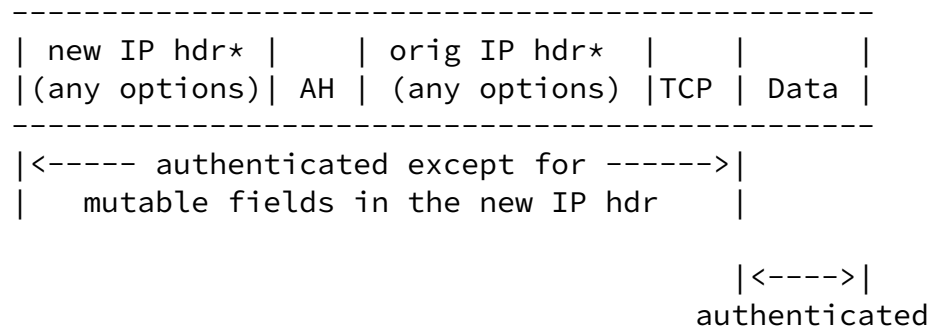
Using the above example on TCP, here are the datagram before applying AH:



and after applying ML-IPSEC AH transport mode:



and after applying ML-IPSEC AH tunnel mode:

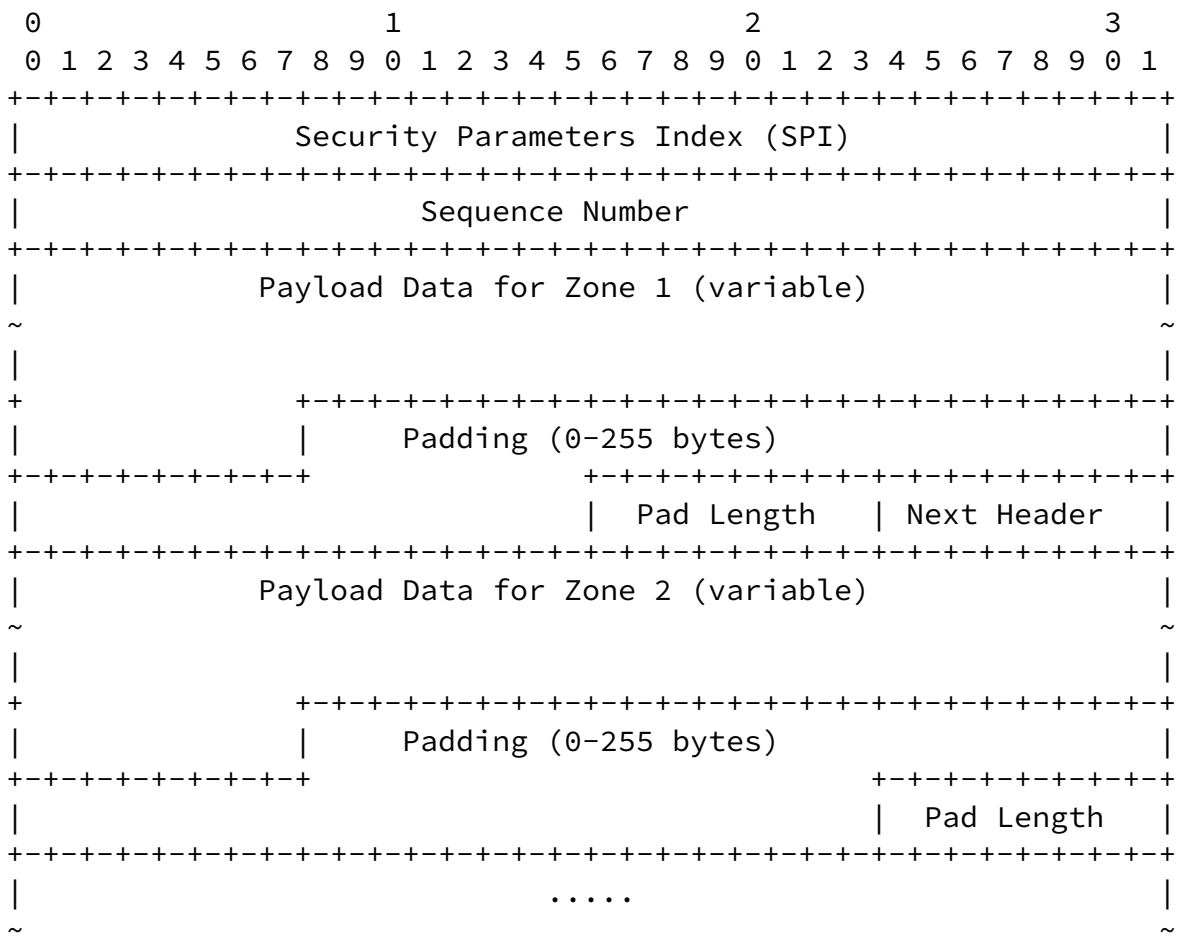


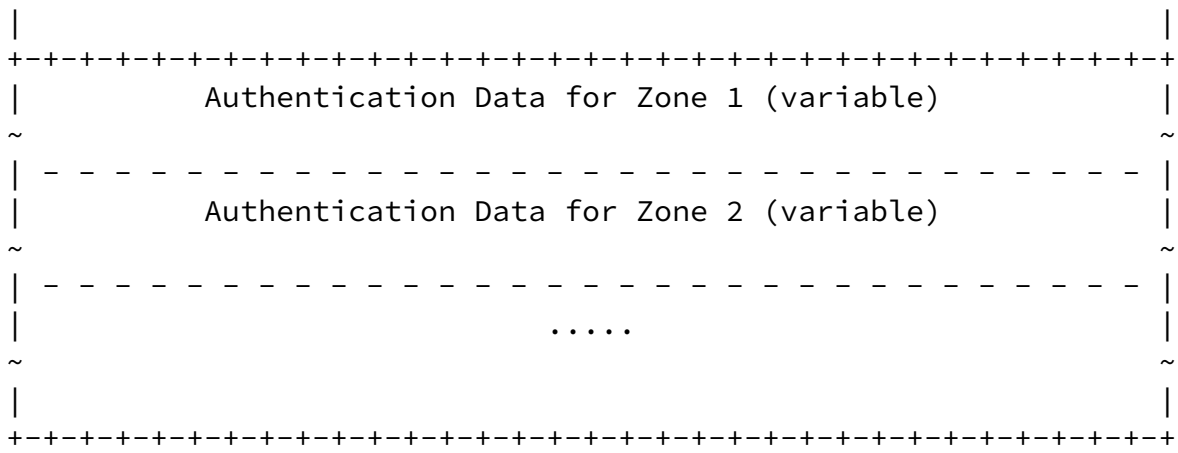
On inbound processing, a zone is authenticated only if the corresponding zonal SA is non-null. The ICVs are calculated the same way as described above, and the values are then matched against the ICVs stored in the Authentication Data. In an intermediate node, a packet will go through inbound processing and then outbound

processing. If changes are made to the packet in an authorized zone, the ICV is recomputed and stored in a proper place in the Authentication Data field. ICV data of unchanged zones are left untouched.

5.3 ESP Headers

ML-IPSEC is perhaps more useful in ESP, where the IP datagram can be encrypted using different keys in different SAs. The following ML-IPSEC ESP header format follows the principle in IPSEC ESP (RFC 2406):



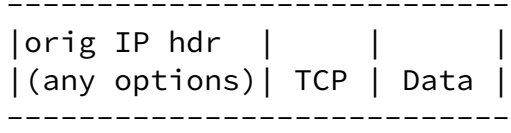


Unlike IPSEC ESP, the Payload Data field in ML-IPSEC ESP are broken into pieces, one for each zone. The Payload Data for each zone, together with Padding, Padding Length, and Next Header field (only in the designed zone), are collectively referred as the ciphertext block for the zone. The size of each ciphertext block can be determined by the CSA, as all zones except the last one are fixed in size.

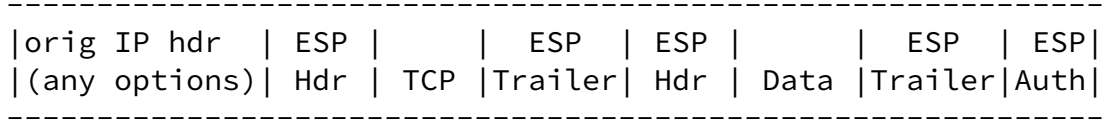
Similar to ML-IPSEC AH, the "Authentication Data" field is a variable-length field that contains several Integrity Check Values (ICVs) for this packet, if authentication has been selected. The

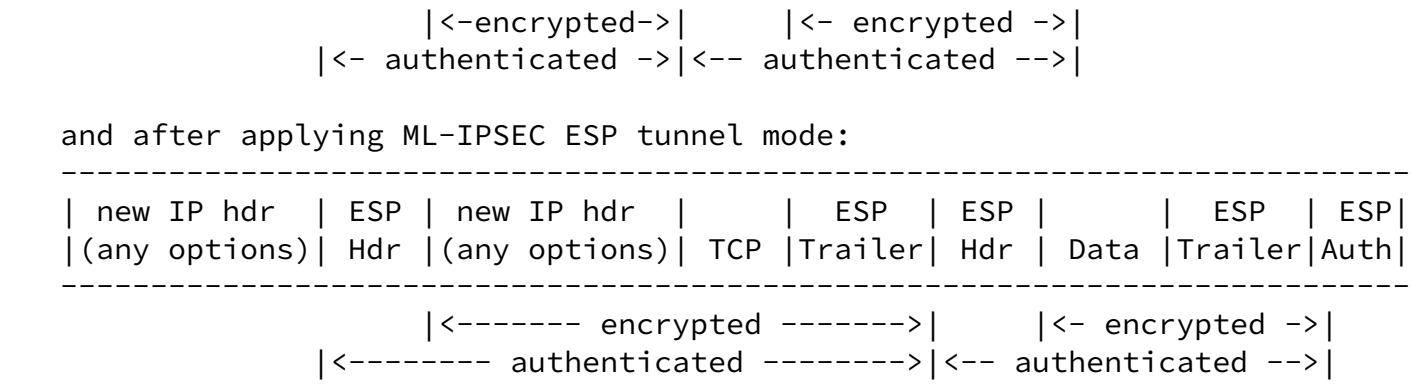
size of each ICV is determined by the authentication algorithm used in each zonal SA, but must be an integral multiple of 32 bits. The boundaries of these zonal authentication data can be derived from the CSA.

Using the above example on TCP, here are the datagram before applying ESP:



and after applying ML-IPSEC ESP transport mode:





5.4 Zone-by-zone Encryption

On outbound processing, the sender takes the following steps in packet encryption:

Step 1, Zone-wise Encapsulation. For each zone, all octets of all subzones are concatenated (in the order they appear in a datagram) and then encapsulted into the ESP Payload Data field for the corresponding zone.

Step 2, Padding. The sender adds any necessary padding to each zone's Payload Data field, to meet encryption algorithm's block size requirement if any, and to align it on a 4-byte boundary.

Step 3, Encryption. The sender then encrypts the result plaintext (Payload Data, Padding, Pad Length, and Next Header) using the key, encryption algorithm, algorithm mode indicated by the zonal SA and cryptographic synchronization data (if any).

After packet encryption, if authentication is selected, the sender computes the ICV from the ciphertext of each zone.

On inbound processing, ML-IPSEC ESP first performs ICV check on a zone-by-zone basis (if authentication is selected). Then, for a zone whose zonal SA is valid and non-null, the receiver decrypts the ESP Payload Data, Padding, Pad Length, and optional Next Header using the key, encryption algorithm, algorithm mode, and cryptographic synchronization data (if any), indicated by the zonal SA. After processing Padding, the receiver then reconstructs the original IP datagram from the original IP header (transport mode) or the tunnel

IP header (tunnel mode), plus the IP payload stored in all the Payload fields. In the reverse procedure of encryption, the receiver take Payload Data of a zone and restore the bytes back according to the zonemap. If a zone has a null SA, the bytes corresponding to the zonemap will be left zero.

In an intermediate node, a packet will go through inbound processing and then outbound processing. If changes are made to the packet in an authorized zone, the receiver will have to re-encrypt the zone and save the ciphertext back to the corresponding Payload Data field.

Disclaimer

The views and specification here are those of the authors and are not necessarily those of their employers. The authors and their employers specifically disclaim responsibility for any problems arising from correct or incorrect implementation or use of this specification.

Author's Address

Yongguang Zhang
HRL Laboratories, LLC
3011 Malibu Canyon Road
Malibu, CA 90265

Phone: (310) 317-5147

EMail: ygz@hrl.com