Multilayer IPSec (ML-IPSec) Protocol Design for improved security performance over satellites

Dr. Haitham Cruickshank and Muhammad Nasir Mumtaz Bhutta
Center for Communication Systems Research, University of Surrey, Guildford, GU2 7XH, UK
John Ashworth and Martin Moseley
EADS Astrium, Portsmouth, UK

Abstract
There are a variety of satellite applications that require application intelligence at intermediate devices for their proper functioning e.g. satellite networks using Performance Enhancing Proxies, PEPs, real time streaming applications like SIP, H.323 and peer-to-peer applications. Interworking between PEPs and security system has been researched in the past. Multi-layer IPSec (ML-IPSec) resolves the conflict between end-to-end security in standard IPSec and working of PEPs. This paper presents the concept and detailed design of ML-IPSec by breaking the IP datagram into three zones while enabling the intermediate nodes to access the TCP header and HTTP header information. The paper also presents an efficient interworking scheme between ML-IPSec and secure IP multicast using the Logical Key Hierarchy for key distribution.

1. Introduction
There are a variety of applications in the networking world today that require application intelligence at intermediate devices for their proper functioning e.g. satellite networks using Performance Enhancing Proxies (PEPs) [1], [2]. There are two common types of PEP: Transport layer (T-PEP) such as TCP PEP and Application layer PEP (A-PEP) such as HTTP accelerators. Also intermediate devices are used in real time streaming applications with SIP, VoIP and peer-to-peer applications such as Napster, Netmeeting etc. Web caching is widely used [3], and implementing PEP will have great impact on their performance especially if satellite and wireless links are used [4].

The security architecture of the Internet Protocol known as IP Security (IPSec) is the most advanced effort in the standardization of Internet security [5]. The IPSec protocol suite is used to provide interoperable cryptographically based security services (i.e. confidentiality, authentication, integrity, and non-repudiation) at the IP layer. It is composed of an authentication protocol [6]: Authentication Header (AH), a confidentiality protocol [7]: Encapsulated Security Payload (ESP) and it also includes an Internet Security Association Establishment and Key Management Protocol (ISAKMP) [8]. These security protocols are designed for both IP version 4 (IPv4) and IP version 6 (IPv6) environments. Transport mode is used only in host-to-host authentication while tunnel mode can be used between two hosts, a host-to-gateway and gateway-to-gateway. The tunnel allows the host to delegate the security service to the gateway. This is especially interesting for companies with two private distant networks connected through the public Internet. IPSec in transport or tunnel mode can impact the operations of PEPs.

Interworking between PEPs and security system has been researched in the past [9]. One solution was the use of an intelligent switch at the PEP. As such, the PEP provides acceleration for the unencrypted packets, while the encrypted packets are allowed to bypass the PEP. With this approach, the applications can choose either security or performance but not both.

Transport Friendly ESP (TF-ESP) or Modified ESP (M-ESP) [9] proposes a modification to ESP header to leave the TCP header information outside the scope of ESP encryption. The mechanism proposes that the unencrypted TCP header information in ESP should be authenticated for integrity. Although this method addresses the performance issues, it exposes enough information to make the connection vulnerable to security threats [10].
Some other solutions explore the use of transport layer security. Secure Socket Layer (SSL) as proposed by Netscape and later been standardized by IETF as Transport Layer Security (TLS) [11] is a transport layer mechanism that provides data security. It encrypts the user data, but not the transport layer headers, such as TCP headers. Since the transport layer headers are in plaintext, the intermediate nodes (such as PEPs) can access or modify them; thereby the performance related issues can be resolved. However, it is not recommended to have TCP headers in plaintext due to security concerns [10]. Suggestions were also made to use SSL/TLS with IPSec in order to protect the header information. The use of SSL/TLS with IPSec is not a good solution because PEP cannot function as IPSec encrypts the TCP headers.

In summary, there is a requirement that security must be implemented in such a way that allows ST and Gateway PEPs to access the transport protocol headers for transport layer PEPs and HTTP content for application layer PEPs.

2. Multi Layer IPSec (ML-IPSec) concept

Multi-Layer IPSec (ML-IPSec) enhances the functionality of IPSec in order to solve the conflicts between IPSec and intermediate entities such as TCP and application layer PEPs. ML-IPSec provides security services as IPSec does with the following requirements taken into consideration:

- ML_IPSec should be fully compatible with original IPSec in format and software processing.
- ML-IPSec should be based on the data structures and building blocks used in IPSec so that it can be easily added to the IPSec implementation.

ML-IPSec concept had been researched in the past [12][13][14]. It was presented to the IPSec WG in the IETF, but it was rejected due to the expected complexity of its security association. Also at that time, IPSec was going through a critical re-definitions of IKEv2 and any extra complexity was seen as unnecessary. The work in this paper, builds on top of the existing research and try to lay the foundation for simpler security association using the multicast key management concept.

First let us have an overview of ML-IPSec. The IP datagram is divided into portions. A portion under the same security protection scheme is called “Zone”. The zones do not overlap. As an example, usually the portion of IP datagram that contains TCP header is zone 1 and the TCP data portion is zone 2.

A zone is also not required to be a continuous block of IP datagram instead it can consist of disconnected blocks within IP datagram. Each disconnected block in a zone is called “Sub-zone”. The zones are formed in terms of octets. A zone map is a mapping relationship from octets of the IP datagram to the associated zones for each octet. The zone boundaries must remain fixed within the lifetime of a security association otherwise it will be very difficult to do zone by zone decryption and authentication. Figure 1 shows an example of a generalized zone map.

Security Association (SA) in IPSec defines the relationship between sender and receiver. The Composite Security Association (CSA) in ML-IPSec also includes the intermediate trusted nodes in addition to the sender and receiver. For each zone, there is an individual security association. Therefore, all security associations for all zones collectively form a CSA to cover the entire IP datagram. A CSA has two elements. The first element is zone map and second element is a zone list. Zone map shows the coverage of each zone in IP datagram and second element, zone list shows the list of SAs for each zone.
The parameters of security associations in IPSec and ML-IPSec are identical except some fields related to zones. The same security protocols AH and ESP are used for ML-IPSec for both transport and tunnel mode with some modifications in some fields of the header.

As shown in Figure 2, the ESP header is also the same as IPSec ESP with some changes. The ESP payload data field in ML-IPSec is divided into multiple pieces depending upon the number of zones. The payload data for each zone collectively along with padding, padding length and next header field is referred to as cipher text block of the zone. In ML-IPSec, different IP datagram parts can be encrypted using different keys for different zones. The ESP authentication data field is also variable in length and contains multiple ICVs which are calculated for different zones and the size of them is dependent on the algorithms being used for integrity.

To show the outbound and inbound processing of an IP datagram, a two-zone example is presented with IPSec security gateways at different ends. The outbound and inbound processing can be referred as processing at sender end and receiver processing at end respectively.

![Figure 2 ML-IPSec ESP Header](image-url)
2.1 Outbound Processing in ML-IPSec

The outbound processing is illustrated in Figure 3.

![Diagram](image_url)

Figure 3 Example of Outbound ML-IPSec Processing

Verification is performed only if the SA is not null for that zone.
For outbound processing the packets passes through the following stages of processing in packet encryption:
- Zone-Wise Encapsulation: For each zone, the octets of all zones are first concatenated and then encapsulated into the ESP payload data field for the corresponding zone.
- Padding: Depending upon the encryption algorithm block size, the payload of each zone is padded to meet the block size requirement.
- Encryption: The resultant payload of above operations is then encrypted according the key, the encryption algorithm and algorithm mode as indicated by zonal SA.

2.2 Inbound Processing in ML-IPSec

The inbound processing in ML-IPSec is reverse of the outbound processing. The processing steps for inbound processing are shown in Figure 4.

In an intermediate node (such as PEPs), a packet will go through partial inbound and then partial outbound processing. The steps of inbound and outbound processing remain same as described above. If the packet is modified during partial inbound processing then outbound processing must redo the authentication and/or encryption for that zone and also should replace the ICVs with new before forwarding to the next hop.
3. ML-IPSec testing Plans

3.1 ML-IPSec scenario and using the MIDCOM architecture

To evaluate the ML-IPSec design, a scenario for different VPNs of a company is considered which are connected via satellite. Nodes in VPN1 communicate with nodes VPN2 and vice versa. The applications of interest here are file transfer and web services with a standard TCP/IP based client/server communications. However and for this testbed evaluation, the IP, TCP and HTTP headers are considered to make them accessible at intermediate nodes. The security gateways deployed on satellite ground stations are ML-IPSec enabled (see Figure 8). For our selected services (e.g. IP, TCP, HTTP), running in middle box as MIDCOM agents, IP datagram is broken into 3 Zones. The satellite gateway internal architecture is shown in Figure 9, focusing only on IP, TCP and HTTP services running as MIDCOM agents (PEPs) [15].
4.2 ML-IPSec Protection Model Details

The ML-IPSec protection model can be defined as follows:
- Zones can be defined for IP header, TCP header, HTTP header and data.
- Different protection schemes for different described Zones (e.g. SA, public/private keys, access control rules etc.) are used to provide security services.

Zones and Zone Map

The Zones and Zone Map for ML-IPSec with accessible HTTP header at intermediate nodes is described in Figure 10. As there can be many services (e.g. TCP PEPs and HTTP PEPs.) running inside an intermediate node according to MIDCOM protocol inside a middle box (intermediate node), as shown in Figure 9. The ML-IPSec protection model can be applied as a general solution to intermediate nodes to break IPSec IP datagram into as multiple Zones as requested by applications in middle box as MIDCOM agents. More details on MIDCOM agents and protocol are available in [15].

Composite Security Association (CSA)

As CSA is contains “Zone Map” and “Zone List” as components. The Zone Map and Zone List are shown below:
- Zone 1 = (1 – 20) + (42 + m) – (42+m+n) Bytes
- Zone 2 = 21 – 40 Bytes
- Zone 3 = 41 – (41+m) Bytes

In Zone List area we show the SAs, their parameters and access control.
4.3 Testbed Use Cases and Measurements

A testbed is being developed and the plan is to consider the following use cases:
- IP Only: running standard IP with no security.
- IPSec: running IPSec using ESP with authentication mode enabled.
- ML-IPSec (1 Zone) = IPSec
- ML-IPSec (2 Zone)
- ML-IPSec (3 Zones)

The ML-IPSec experiment will be evaluated for processing delays, CPU overload and bandwidth overhead.

**Processing Delay**
The processing delay will be measured by taking following parameters into consideration:

- One Host pinging other
- Repeated runs of simulations will run.
- Packet size will be fixed.
- Processing Time will be evaluated.

**Comparing CPU Overload**
For evaluation of CPU overhead environment will be configured as given below:

- One host generate and send packets as fast as it can and other counting after receiving.
- CPU speed will be fixed.
- Network speed will be fixed.
- Network Throughput = CPU load in sending and receiving packet as selected network speed will be much higher than CPU generating capacity.
- Throughput and CPU load relationship will be studied.

**Bandwidth Overhead**
For bandwidth overhead evaluation following experiment will be conducted:

- One host generate and send packets as fast as it can and other counting after receiving.
- CPU speed will be fixed.
- Network speed will be fixed.
- The network speed will be much low than the CPU packet generating capacity.
- Throughput and protocol overhead relationship will be studied.

5. Conclusion

The ML-IPSec can solve the interworking issues between intermediate devices such as PEPs and IPSec. ML-IPSec can enable PEPs to access a limited portion of IP datagram for their proper functioning while end-to-end data confidentiality is preserved by ML-IPSec. Such performance improvement will have a strong impact on future web caching over satellite and other wireless networks.
The paper presented ML-IPSec overview and its design detailed by breaking the IP datagram into three zones while enabling PEPs to access the TCP header and HTTP header information. The paper presented the inbound and outbound processing of the composite security association. Also it analyzed the complexity of ML-IPSec when applied to an IP datagram broken into 3 zones with different security schemes. A testbed is being developed and the ML-IPSec experiment will be evaluated for processing delays, CPU overload and bandwidth overhead.

References