A New Dynamic Multilayer IPSec Protocol

Muhammad Nasir Mumtaz Bhutta and Haitham Cruickshank

CCSR, University of Surrey
Guildford, Surrey, United Kingdom
{m.bhutta,h.cruickshank}@surrey.ac.uk

Abstract. Performance Enhancing Proxies (PEPs) are used in satellite networks for better performance of the TCP/IP applications. Multi-layer IPSec (ML-IPSec) resolves the conflict between end-to-end security in standard IPSec and operation of PEPs. Previous Ml-IPSec solution has issues of limited application scope and increased complexity to implement and process the ML-IPSec protected data. This paper presents a new dynamic ML-IPSec protocol which addresses these issues. The paper also analyzes the protocol with reference to previous ML-IPSec protocol and presents the experiment performed to analyze the network performance while running IPSec and ML-IPSec.

Keywords: ML-IPSec, IPSec, PEP, TCP, Dynamic ML-IPSec.

1 Introduction

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. More information on Y. Zhang work on ML-IPSec can be obtained in [1], [2]. The earlier work on ML-IPSec, done by HRL Laboratories, was presented to IETF in many meetings and an internet-draft was written as well. IETF showed concern in three areas: 1) the idea presented by HRL Laboratories was only targeting very limited domain by fixing the zone map for the security association lifetime, 2) implementation complexity was increased and 3) it was required to show two more actual implementations of ML-IPSec. However, the problem of complexity of key management and security association setup for intermediate devices is also very complex and costly operation in terms of communication and it is not addressed very well. The HRL Laboratories suggested using "Internet Key Exchange (IKE v2)" for key setup. For large networks with large number of intermediate devices, using IKE v2 is not a good option. Also there are requirements for changing the databases of IPSec and IKE to make it compatible with ML-IPSec. The ML-IPSec analysis, design and IETF issues are discussed in detail by M.Bhutta and H.Cruickshank in [3], [4]. The issues are solved by our proposed novel, new dynamic ML-IPSec protocol. The paper also describes in detail the new proposed Dynamic ML-IPSec design and proof of study performed on SSFNet simulator.

P. Pillai, R. Shorey, and E. Ferro (Eds.): PSATS 2012, LNICST 52, pp. 119–129, 2013.

2 Previous Multilayer IPSec

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". 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.

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.

As Encapsulating Security Payload (ESP) in IPSec provides the maximum security features of IPSec protocol, so here our focus is on ESP only. The discussion on Authentication Header (AH) is out of scope of this paper.

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. More information on previous ML-IPsec can be found in [1], [2].

3 Issues in Previous ML-IPSec

As notified by IETF, the application scope and increased implementation complexity are main issues in previous ML-IPSec [5, HRL Laboratory report]. Also, key management for ML-IPSec is a very complex and big concern to make ML-IPSec enable to provide security services. Y. Zhang proposed to use IKEv2 to establish the security associations between intermediate communicating parties but, using IKEv2 will not scale well and complexity will also increase as network will grow. However, the main focus here is to address the limited application scope of the previous ML-IPSec protocol. The key management complexity is out of the scope of this paper.

The main reason for limited application scope of previous ML-IPSec protocol is the way how zone map is established. As described earlier in section II, zone map defines the coverage of each zone in IP datagram. The zone map is part of composite security association (CSA) and is established between communicating parties when security association (SA) is established. The zone map must remain constant for the duration of established security association (SA) life time. By making zone map

constant, restricts many applications to use ML-IPSec like HTTP in which request/response size is variable due to appending cookies etc or any application appending extra headers will also not work for constant zone map. This paper presents a new dynamic ML-IPSec protocol which increases the scope of application and address the constant zone map problem.

4 New Dynamic ML-IPSec Protocol

The proposed new dynamic ML-IPSec protocol increases the application support by allowing breaking down the IP datagram into zones as per requirement. Before describing the details of new ML-IPSec protocol, we summaries the design considerations here. We propose that zone map should not be part of CSA; instead the zone information is embedded in the ESP header. CSA will remain same except the zone map information. It will contain the zone list and all designated and non-designated security association parameters will also remain as part of CSA and will be described in the same way. The zone information about the IP datagram will go as part of ESP.

The inbound, outbound processing on participating nodes, ESP and AH packets parsing and security processing on ESP and AH are affected by making zone information part of ESP and AH header. The paper only focuses on ESP but, the basic logic and processing with respect to zones will be same for AH as well. All these processing procedures and design details are described in this section.

4.1 Zones and Zone Map

A zone in new ML-IPSec is a continuous block (portion) of IP datagram. The reason to identify a zone as a continuous block is due to the reason as it reduces the complexity to process a part of IP datagram. The detailed discussion on zone manipulations is in the coming sub-section, Security Protocols, where we discuss all the processing details.

A zone map in previous ML-IPSec was described as bit-by-bit mapping of IP datagram into zones. However, in new ML-IPSec protocol zone map also contains information about zones but, zone map is combinations of zone pointers. A zone pointer points to the starting bit location in ESP header when IP datagram is encapsulated after encryption. Further details about zone map are described in sub-section, Security Protocols, along with parsing and processing details of ESP header and IP datagram.

4.2 Composite Security Association (CSA)

CSA in previous ML-IPSec consists of two elements, zone list and zone map. Zone list is a list of all security associations associated with each zone and zone map defines the IP datagram bits associated with each zone. However, CSA only consists of zone list without zone map as zone map information becomes part of ESP header. Except this change, CSA remains unchanged including the definitions of designated and non-designated parameters.

4.3 Security Protocol

As described in section II.C, the paper focuses only to discuss security protocols with respect to ESP only.

The figure 5 shows the new ESP header. The ESP header is also the same as IPSec and previous ML-IPSec ESP header with some changes.

4.3.1 ESP Header Design

The ESP header contains SPI and sequence number as in IPSec and previous ML-IPSec. After sequence number field, the ESP header contains a 4 bit field called "Total Zones". This field identifies the total number of zones of each IP datagram under process. The IP datagram can be broken into as many zones as required by application up to a maximum of 15 zones.

After "Total Zones" field, there are variable numbers of pointers fields depending upon the number of zones. Each pointer field consists of 11 bits to support the IP MTU of 1400 bytes and points to the starting position of each zone inside ESP payload. After adding all the pointer fields, the necessary padding is done to keep ESP header word size constant of 32 bits. The "Pad Length" 5 bits field describes the total number of padding bits added in the ESP header for pointers.

The ESP payload data in new ML-IPSec protocol consists of multiple pieces depending upon the number zone. The data in each zone is encrypted after adding any necessary padding and then ICV is calculated on the resultant cipher data for that zone. The cipher data and ICV for a specific zone is combined and put in the ESP payload data at specific position. The resultant encrypted data and ICV of different zones are encapsulated in the ESP payload in the order consistent with IP datagram data.

The figure 5 shows the new ESP header for n number of zones.

4.3.2 ESP Header Processing

The ESP header processing at participating nodes is in consistence with previous ML-IPSec and IPSec protocols. The processing steps described here are followed in outbound and inbound processing performed in participating entities in the communication. The processing steps are as follows:

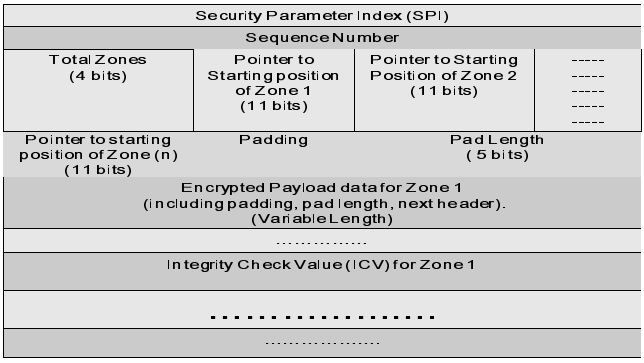

Fig. 1. ESP Header for New ML-IPSec Protocol

- For encryption, the payload data part for each zone is appended with necessary padding and pad length depending upon the algorithm and then encrypted. However, the designated zone also contains the "Next Header" field as in previous ML-IPSec. For decryption operation, the decryption is done depending upon the algorithm selected and original data is received after truncating the padding and pad length fields. The operation steps performed for encryption/decryption are in consistence with IPSec and ML-IPSec.
- The ICV calculation steps in new ML-IPSec are same as in IPSec and previous ML-IPSec depending upon the selected algorithm. However, the way ICV for each zone is encapsulated in ESP header is different from previous ML-IPSec. In previous ML-IPSec ICVs for all zones were combined together and then appended at the end of combined encrypted data of all zones but, in new ML-IPSec ICV for each zone is appended at the end of encrypted data for that specific zone to make it easy for processing and less complex. As ICV is always calculated to fixed size depending upon the algorithm selected, it can be easily extracted from the end of zone encrypted data at receiving end to verify the integrity.

4.4 Inbound, Outbound Processing

The outbound and inbound processing can be referred as processing at sender end and receiver processing at end respectively. In intermediate nodes, partial processing is done on IP datagram accessible part; however the steps remain same with respect to inbound and outbound processing.

4.5 Outbound Processing in ML-IPSec

In new ML-IPSec protocol the outbound processing is done in the same way as described in section II.D with the following exceptions:

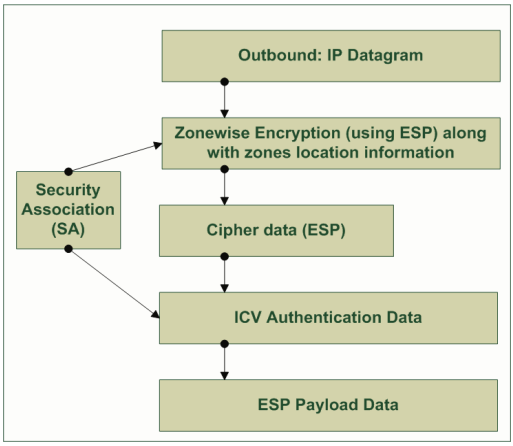

Fig. 2. Example of Outbound Processing

- There is no pre-defined constant zone map which is consulted to perform security operations on each selected zone. The zones are generated dynamically and are processed for encryption and integrity check value calculation. However, the sequence of performing security operations remains same.

The processing steps for outbound processing are shown in figure 6.

4.6 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 7:

5 Security Services Offered by New ML-IPSec Protocol

The new ML-IPSec offers same security services as offered by IPSec and ML-IPSec including origin authentication, connectionless integrity, optional partial sequence integrity, data confidentiality and limited traffic flow confidentiality with the help of ESP.

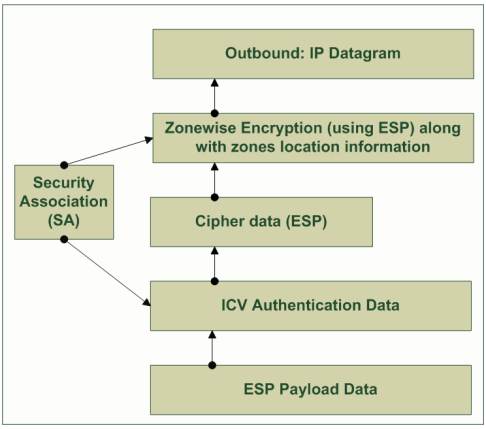

Fig. 3. Example of Inbound Processing

6 Performance Evaluation of New Dynamic ML-IPSec Protocol

To evaluate the performance of new proposed dynamic ML-IPSec, we selected to modify the implementation of IPSec by NIST. The simulator used by NIST was SSF/SSFNet and was developed in Java. For proof of study and analyze the network performance while running Dynamic ML-IPSec, we have modified the NIST IPSec implementation according to our proposed design. Following are the details of experiment performed.

6.1 Experiment Environment

A network of asymmetric kind where one part of network only contained clients and one part of network only contained servers were arranged in dumbbell topology as shown in figure 8. The network consisted of a pair of security gateways and one or more hosts behind each gateway (connected with a LAN). Each gateway provides secure VPN services using Dynamic ML-IPSec and IPSec for the local hosts and acts as the SA initiator or the SA responder. All the experiments were performed in tunnel mode where a security policy controls the packets to be processed. For our experiment, we configured the host behind security gateway to create a security association with each other node.

In the experiment, we used manual key management. So at start of experiment, a SA was established for IKE and for IPSec/ML-IPSec. In ML-IPSec experiment, the SAs were different for different zones. The life time of SA was more than the experiment time, so that no re-keying should be required for life time of experiment.

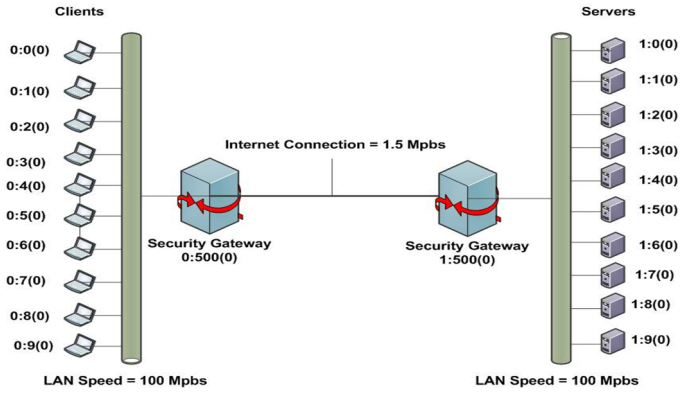

Fig. 4. A sample of network configuration used for experiment

6.2 Traffic Models

In the simulation, different experiments were performed for file transfer between TCP clients and TCP servers. A client connects to a randomly chosen server and requests server to transfer a file of fixed size for the selected duration of experiment. A TCP-based application continuously generates fixed number of files traffic for each session for the duration of the experiment. The TCP clients were waiting for a random time to send the next request after completion of the session.

6.3 Performance Measures

The purpose of the study was proof of concept of dynamic ML-IPSec that it functions according to the policy and we achieved similar behavior between IPSec and

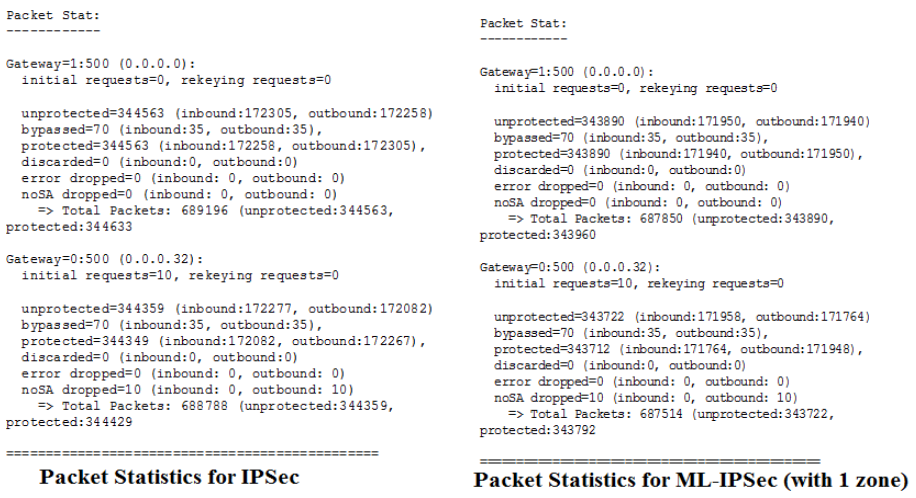

Fig. 5. Example of Inbound Processing

ML-IPSec for same set of security related experiments. We also tried to compare the network performance while running IPSec and ML-IPSec for different traffic loads and different network bandwidth configurations.

The packet statistics shown in figure 9 are counted according to the security policy enforcement and selected network conditions. We counted the protected/unprotected/bypassed/error packets for both inbound and outbound independently within a security gateway.

For the application layer, we gathered the session throughput for different network conditions. We run the simulation in tunnel mode for fixed time. The internet connection between securities gateways were configured for different bandwidth values from 1 Mbps to 10 Mbps to see the effects of network bandwidth with respect to traffic load. To change the traffic load, we changed the file sizes from 1 Mbytes to 20 Mbytes. The security attributes like 3DES_CBC, AES or HMAC_SHA1 etc, can change the overall performance under specific environment. We considered the affects of our selected security schemes to play a role in our analysis. The cryptographic figures used for our experiment are given below and are in consistence with NIST IPSec experiment.

Table 1. Cryptographic Figures used for experiment

Algorithms	Block Size	Key Size
3_ DES_CBC	8 bytes	24 bytes
HMAC_SHA1	64 bytes	20 bytes

As shown in figures 10 and 11, overall throughput of network becomes stable once the network can handle all the transmitted data efficiently than the speed data is transmitted on the network. There remains unnoticeable difference between the performance while running IPSec and ML-IPSec with different number of zones. However, when network speed is less than the speed at which processor is transmitting the data on the network; we have observed that configuration where less processing is involved gives better performance as compared to configuration where high processing is required. Hence in low bandwidth network, obviously IPSec gives better performance as compared to ML-IPSec. However, this difference is not very high. The below graph and data table shows the average throughput obtained in the analysis.

Table 2. Network Throughput for file size of 20MBytes

File Size (MB)	Bandwidth (Mbps)	Average Throughput			
		IPSec	ML-IPSec (1 Zone)	ML-IPSec (2 Zones)	ML-IPSec (3 Zones)
20	1	92.842	92.669	91.906	91.822
20	2	185.095	184.74	183.235	183.07
20	3	251.796	251.793	251.776	251.776
20	4	252.02	252.02	252.01	252.006
20	5	252.13	252.13	252.13	252.13
20	6	252.23	252.23	252.22	252.22
20	7	252.286	252.286	252.286	252.283
20	8	252.336	252.333	252.33	252.33
20	9	252.336	252.336	252.33	252.33
20	10	252.403	252.4	252.396	252.396

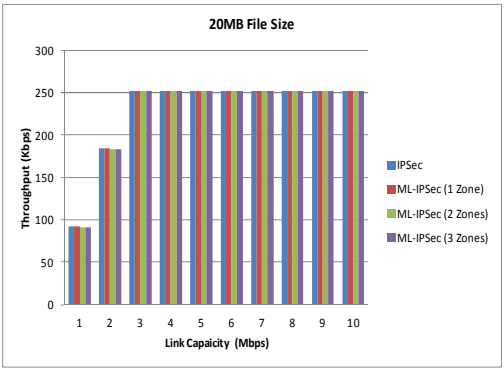

Fig. 6. Network Throughput for file size of 20MBytes

The figure also shows that as file size increases, the TCP application gives better utilize the network bandwidth as compared to low file sizes. The low file size degrades the network performance to very small extent.

6.4 Limitations of Implementation

The analysis was performed using SSF/SSFNet simulator and NIST IPsec implementation. However, for our analysis there are some limitations inherited from NIST IPSec implementation and SSF/SSFNet implementation which are given below:

- No actual implementation of cryptographic algorithms, keys and cryptographic operations is done. However, the security processing behavior was simulated alongside processing block size and processing time.
- The header sizes and data sizes may be different from actual implementations for different constraints of Java language.

However, the overall behavior of IPSec, ML-IPSec will not be some much different from real implementation.

7 Conclusion

The ML-IPSec can solve the interworking issues between intermediate devices such as PEPs and IPSec. ML-IPSec enables PEPs to access a limited portion of IP datagram for their proper functioning while end-to-end data confidentiality is preserved by ML-IPSec. However, there are some issues in previous ML-IPSec solution like limited application domain, which can be resolved by new dynamic ML-IPSec by making application more flexible to break IP datagram into different zones. The new dynamic ML-IPSec also improves the efficiency and reduces complexity to encapsulate the zones information into ESP payload.

The paper presented new dynamic ML-IPSec with detailed description of its design and processing. The paper has also performed an analysis on new dynamic ML-IPSec in comparison with IPSec and previous ML-IPSec where appropriate. The paper has shown some results of our analysis for security policy enforcement and network performance evaluation with different network bandwidth and traffic load. It is observed that ML-IPSec gives almost same performance as IPSec performs when network bandwidth is more than 3 Mbps. However, when network bandwidth is low, then there is small performance reduction as compared to IPSec.

References

[1] Zhang, Y.: A Multilayer IP Security Protocol for TCP Performance Enhancement in Wireless Networks. IEEE Journals on Selected Areas in Communicaitons 22(4) (May 2004)

[2] Zhang, Y., Singh, B.: A multi-layer IPsec protocol. In: Proc. Usenix Security Symp., pp. 213–228 (August 2000)

[3] Cruickshank, H., Bhutta, M.N.M., Ashworth, J., Moseley, M.: Interworking between Satellite Performance Enhancing Proxies and Multilayer IPSec (ML-IPSec). In: 16th KA and Broadband Communications 2010, Milan, Italy (2010)

[4] Bhutta, M.N.M., Haitham, Ashworth, J., Moseley, M.: Multilayer IPSec (ML-IPSec) Design for Transport and Application Layer Satellite Performance Enhancing Proxies. In: 28th AIAA International Communications Satellite Systems, AIAA/ICSSC, Anaheim, California (2010)
[5] Zhang, Y.: HRL Laboratories Report, Multi-layer Internet Security for Satellite and Wireless Networks (December 1999)
[6] Border, J., et al.: Performance Enhancing Proxies Intended to Mitigate Link-Related Degradations. IETF RFC 3135 (June 2001)
[7] Cruickshank, H.: Technical Report on Performance Enhancing Proxies (PEPs) for the European ETSI Broadband Satellite Multimedia (BSM) working group. ETSI Report TR 102 676 (September 2009), http://portal.etsi.org
[8] Gomez, C., et al.: Web browsing optimization over 2.5G and 3G: end-to-end mechanisms vs. usage of performance enhancing proxies. Wireless Communications and Mobile Computing 8, 213–230 (2008)
[9] Kent, S., Seo, K.: BBN Technologies, Security Architecture for Internet Protocol. RFC 4301 (December 2005)
[10] Kent, S.: BBN Technologies, IP Authentication Header (AH). RFC 4302 (December 2005)
[11] Kent, S.: BBN Technologies, IP Encapsulating Security Payload (ESP). RFC 4303 (December 2005)
[12] Kaufman, C.: Microsoft, Internet Key Exchange (IKEv2) Protocol. RFC 4306 (December 2005)
[13] Obanaik, V.: Secure performance enhancing proxy: To ensure end-to-end security and enhance TCP performance over IPv6 wireless networks. Elsevier Computer Networks 50, 2225–2238 (2006)
[14] Bellovin, S.: Probable plaintext cryptanalysis of the IPSecurity protocols. In: Proceedings of the Symposium on Network and Distributed System Security (February 1997)
[15] Dierks, T., et al.: The TLS Protocol Version 1.2, RFC 5246 (August 2008)
[16] Sing, J., Soh, B.: A Critical Analysis of Multi-layer IP Security Protocol. In: Third International Conference on Information Technology and Applications, ICITA 2005 (2005)
[17] Annoni, M., Boiero, G., Salis, N., Cruickshank, H.S., Howarth, M.P., Sun, Z.: Interworking between multi-layer IPSEC and Secure multicast services over GEO satellites. Eur. Cooperation in the Field of Sci. Tech. Res., Tech. Rep. COST 272 TD-02–016 (2002)
[18] Annoni, M., Boiero, G., Salis, N.: Security issues in the BRAHMS system. In: Proc. Ist MobileWireless Telecommunications Summit 2002 (June 2002)
[19] Baugher, M., et al.: Multicast Security (MSEC) Group Key Management Architecture. IETF RFC 4046 (April 2005)
[20] Cruickshank, H.: Technical Specifications for satellite networks multicast security architecture and key management for the European ETSI Broadband Satellite Multimedia (BSM) working group. ETSI Specifications. ETSI TS 102 466 (December 2006), http://portal.etsi.org
[21] Wallner, D., et al.: Key Management for Multicast: Issues and Architectures. IETF RFC 2627 (June 1999)
[22] Sirsuresh, P., et al.: Middlebox Communication Architecture and Framework. IETF RFC 3303 (August 2002)