Broadband Satellite Multimedia (BSM) Security Architecture and Interworking with Performance Enhancing Proxies

H. Cruickshank[1], R. Mort[2], and M. Berioli[3]

[1] CCSR, University of Surrey, Guildford, UK
[2] Systek, Havant, UK
[3] German aerospace center (DLR), Munich, Germany

Abstract. Satellites had been successful in the past due to their wide area coverage and speedy deployment of new services especially in remote regions of Europe and the rest of the world. The future development of broadband satellite systems providing services based on the Internet Protocol (IP) needs to be stimulated by means of common standards. This paper presents the ETSI BSM PEP terminal architecture, PEP usage scenarios and security configurations for successful PEP implementations.

Keywords: PEP, TLS, IPsec, SI-SAP, TCP and BSM.

1 Introduction

Satellites have been successful in providing infrastructure for broadband telecommunications due to their wide area coverage and ability to speedily deploy new services especially in remote regions of the world. The future development of broadband satellite systems providing IP-based services needs to be stimulated through common approaches and standards where possible. The BSM work is focussed on the efficient transport of IP data streams and on how to interoperate resulting satellite networks with terrestrial IP networks. The BSM standards are being designed to use existing standards (such as DVB-RCS [1]) while remaining open to emerging standards and other available technologies (the ultimate choice is left to the market). This paper presents the ETSI BSM Performance Enhancing Proxy (PEP) architecture, which includes the satellite terminal protocol stack, PEP usage scenarios and security configurations suitable for PEP deployment.

In general, the Internet transport protocol (namely TCP) exhibits suboptimal performance due to the following satellite characteristics:

- Long feedback loops: Propagation delay from a sender to a receiver in a geosynchronous satellite network can range from 240 to 280 milliseconds. This will cause slow connection setup, slow to respond to loss and slow discovery of available bandwidth.
- Large bandwidth*delay products: TCP needs to keep a large number of packets "in flight" in order to fully utilize the satellite link.
- Asymmetric capacity: The return link capacity for carrying ACKs can have a significant impact on TCP performance.

K. Sithamparanathan (Ed.): Psats 2009, LNICST 15, pp. 132–142, 2009.

One solution is implementing end-to-end improvements techniques to TCP and HTTP. However, servers are by default unaware of the access technology used by a client. Therefore, optimizing communications for each particular last hop technology is not possible. In addition, server design principles attempt to optimize server performance rather than user experience. Thus, end-to-end techniques can provide some improvement, but cannot ensure that the best possible improvements. Therefore, these techniques are not the focus of this paper.

Another solution is to place an entity called Performance Enhancing Proxy (PEP) somewhere between the endpoints of a communication link. Among the TCP PEP proposals, one solution is represented by the splitting approach [2]. The rationale of the splitting concept is to separate the satellite portion from the rest of the network. This approach can be further be divided into two categories: Distributed PEPs where the PEP client and server are located at both ends of the satellite link. The other category is integrated PEPs with only one PEP entity residing with the satellite gateway. Typical TCP PEP improvements are:

- TCP Spoofing: Eliminates effects of satellite delay on TCPs slow start and window sizing.
- ACK Reduction: Reduces unnecessary acknowledgements to improve bandwidth efficiency.
- Flow Control: Employs network feedback to intelligently control traffic flow.
- Error Recovery: Works closely with Flow Control to recover damaged or lost packets.
- Traffic Prioritization: Classifies traffic by IP address and port and prioritizes accordingly.
- Connection Establishment Spoofing: Intelligently spoofs the TCP three-way handshake to speed up establishment of a connection.

In addition to TCP PEPs, there are other complementary solutions such as application layer PEPs, where web browsing is the major target for application PEPs. Typical application layer PEPs improvements are:

- HTTP pre-fetching: Intercepting requested Web pages, identifying Web objects referred to by the Web pages, downloading these objects in anticipation of the next user requests.
- Browser Cache Leveraging: Cache’s some web pages not residing in browser cache, improving efficiency.
- Bulk Transfer Prioritization: Prioritizes bulk transfers to prevent adverse effect on other Web traffic.
- Cookie Handling: Ensures accurate painting of Web pages with the proper cookies.
- Compression: Payload compression provides increased transmission speeds. In addition, header compression for TCP, UDP, and RTP protocols results in additional bandwidth savings.
- DNS caching techniques, to further improve bandwidth utilization.

Commercial PEPs normally combined some/all the TCP and application layer techniques together such the XipLink [3], FastSat [4] and Hughes [5] PEPs.

The BSM architecture [6] provides a generic BSM protocol stack for IP services in Satellite Terminals (ST) and Gateways (GW). An important feature is the Satellite Independent Service Access Point interface or SI-SAP interface. This interface provides the BSM with a layer of abstraction for the lower layer functions. It allows the BSM protocols developed in the satellite independent layer to perform over any BSM family (specific satellite technologies). Moreover, the SI-SAP also enables the use of standard Internet protocols for example address resolution, QoS, security and network management, directly over the BSM or with minimal adaptation to BSM physical characteristics. Finally the SI-SAP even makes it possible to envisage switching from one satellite system to another and to even a non-satellite technology while preserving the BSM operator's investment in layer 3 software developments. The aim of the current work on PEPs in the Specific Task Force (STF) 344 in ETSI BSM is to describe the current solutions for PEPs in broadband multimedia satellite systems. The range of PEPs considered includes TCP accelerators, TCP header compression and HTTP proxies. The PEPs are classified in terms of ease of implementation, interworking capability with other PEPs and performance potential. The work also includes an analysis of the various PEP types/mechanisms and recommendations for the use of these PEPs in BSM networks. If the PEP design adopts a satellite-independent approach it can be used with different lower layers without requirement significant redevelopment. This has benefits for both the PEP manufacturers (by reducing the new costs and time of new developments) and also for the end-user who can migrate to a new satellite system while retaining the same or similar “known” PEP properties.

This paper is organized as follows: Section II provides an overview of BSM PEP architecture. Section III presents the past research in security systems related to PEPs. Section IV presents security solutions for BSM PEP architecture. Section V describes the detailed BAM architecture for link layer security with PEPs. Finally section VI concludes this work with a summary and outlook on related work at ETSM BSM group.

2 Overview of BSM PEP Terminal Architecture and Components

A. BSM ST and Gateway components

Figure 1 shows the combined PEP protocol stack with the BSM ST architecture. The PEP client residing with the BSM ST is called ST PEP. On the satellite network side, the ST PEP is connected to BSM ST through an Ethernet LAN. On the terrestrial network side, the ST PEP connects to hosts also in a LAN configuration.

Similarly the PEP server (called Gateway PEP) resides with the BSM gateway. The Gateway PEP has the same architecture to the ST PEP with two interfaces, one to the BSM satellite network and one to terrestrial networks. On the terrestrial networks side, the gateway PEP connects to a content server through the general Internet. Also in many configurations, the Gateway PEP will be located remotely from the BSM Gateway terminal (e.g. Gateway PEP run by an Internet service provider). More detailed on the architecture are presented in section IV.

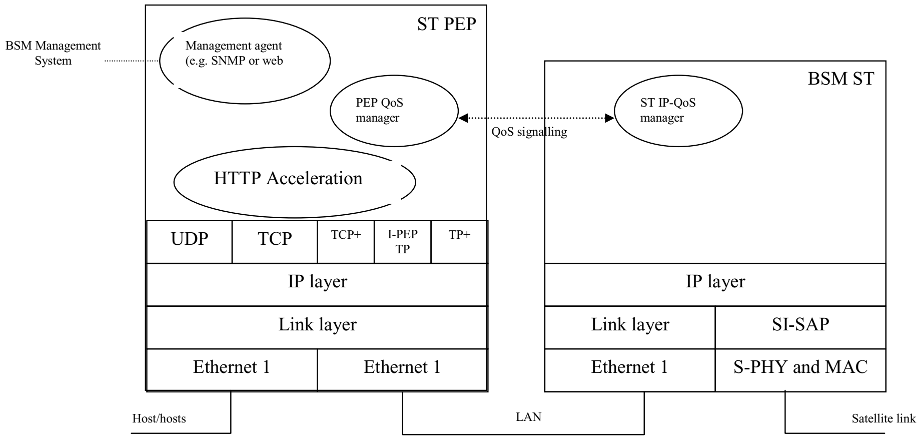

Fig. 1. BSM ST PEP

As shown in Figure 1, the transport protocol in the PEP is divided between standard TCP/UDP and PEP specific transport protocols. The PEP-specific transport protocol can be:

- A modified TCP (TCP+, such as the Hypla protocol [7]), which is used in integrated PEP configurations, where only Gateway PEP will be used (no ST PEP).
- Interoperable PEP (I-PEP) Transport Protocol (I-PEP TP) [8]. This standard protocol recommended by the I-Labs and used in distributed PEP configurations. The I-PEP TP is based on an extension to TCP termed SCPS-TP, which was produced by the Consultative Committee for Space Data Systems (CCSDS).
- Proprietary distributed Transport Protocol (TP+), where other non-standard (company specific) protocols are used.

In addition and as shown in Figure 1, The ST (or Gateway) PEPs can be managed either locally or remotely. For remote management, either SNMP or HTTP protocols can be used to communicate with the BSM management system. In both cases the PEP monitoring and configuration controls can be based on the standard MIB II and enterprise specific PEP MIBs. Also QoS signalling is needed between the ST PEP and the BSM QoS manager in the ST. Such signalling is necessary for QoS monitoring of the ST queues and adjusting rate control parameters accordingly to maximize the use of the satellite capacity. Similar signalling is needed between Gateway PEP and the BSM QoS manager in the BSM Gateway. The optimum PEP performance is expected to require a close matching between the PEP configuration and QoS of the associated lower layer bearer services. This signalling can be based on IntServ or DiffServ architectures [9] and [10] and may require cross layer signalling via the SI-SAP interface. However, the focus of this paper is on security issues and the network management and QoS issues will not be elaborated any further.

B. *A typical PEP scenario*

Figure 2 shows a typical PEP usage scenario with a single user behind the ST PEP (PEP client). This reflects the typical home user or home office configuration. The PEP client may be integrated with the BSM ST, or it may be a stand-alone entity separate from both the end user's device and the ST.

There can be several variations to this scenario. One variation is a multi-user scenario where the same ST PEP serves multiple users in a LAN configuration. Another variation to Figure 2 is where the Gateway PEP (PEP server) is external to the BSM Gateway (satellite terminal), motivating two different set-ups:

- PEP server may be run by a separate Internet Service Provider (ISP) on behalf of many users or
- PEP server may be operated by an enterprise on its own behalf.

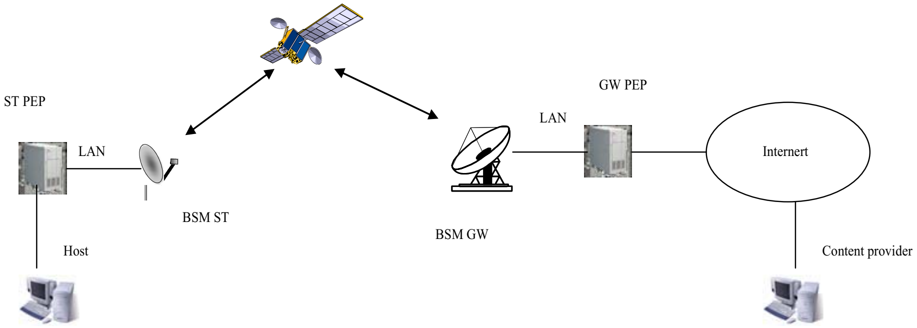

Fig. 2. A typical BSM PEP scenario

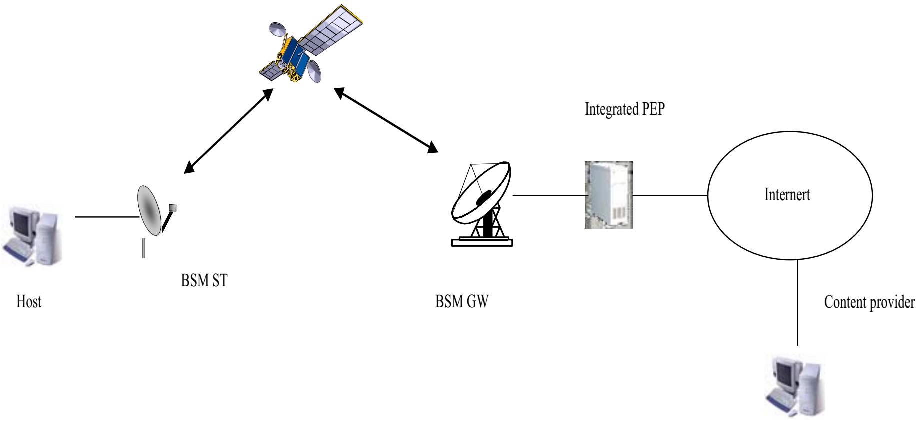

Fig. 3. Integrated PEP implemented at the BSM Gateway

A third variation is the use of multiple Gateway PEPs. The motivation here can be the presence of multiple ISPs or because performance enhancement is managed directly between user sites (VPN configuration). Here the ST PEP needs to interoperate with multiple Gateway PEPs from different vendors. This is an ideal setup for using the I-PEP protocol [8] mentioned earlier.

The previous examples showed various aspects of a distributed PEP (PEP client and server at each end of the satellite link). Figure 3 shows an example of an integrated PEP. The integrated PEP is located only at the BSM Gateway.

Here the TCP connection established among end hosts, is split in two separate connections, with the integrated PEP located at the BSM Gateway. The first connection (between the web sever and the integrated PEP makes use of the TCP standard and is terminated at the PEP. The second connection, between PEP and the final user, can exploit an enhanced TCP version compatible with a standard TCP receiver (such as the Hypla protocols [7] mentioned in section III). In comparison to a distributed PEP, integrated PEP is simpler to use but has limited enhancement capabilities.

3 Previous Research Work Related to PEP Security

Interworking between PEPs and security system has been researched in the past [11]. For example, many researchers had addressed the issue of interworking between IPsec and PEPs. One solution was the use of an intelligent switch at the PEP. As such, the PEP provide acceleration for the unencrypted packets, while the encrypted packets are allowed to bypass the PEP. With this approach, the applications can choose between security and performance, but both are not obtainable together.

Transport Friendly ESP (TF-ESP) or Modified ESP (M-ESP) [11] proposes a modification to ESP header to accommodate the necessary TCP header information in the ESP header outside the scope of 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 [12].

The Multilayer IPSEC Protocol (ML-IPSEC) [11] proposes a multi-layer encryption scheme. The IP datagram payload is divided into zones; each zone has its own security associations and protection mechanisms. For instance, the TCP data part can be a zone, using end-to-end encryption with the keys only shared between end-hosts. The TCP header could be another zone with security associations between the source, destination and a few trusted nodes (such as PEPs). The trusted nodes can decrypt the transport layer headers to provide the performance enhancements. This mechanism ensures security and can accommodate existing performance solutions. Though the requirements are satisfied, the complexity involved is tremendous. Also, the assumption that intermediate nodes are trustworthy may not be acceptable for users preferring end-to-end security.

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) [13], 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 (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 [12]. 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 away that allows ST and Gateway PEPs to access the transport protocol headers (such as TCP). The most negative implication of using PEPs is breaking the end-to-end semantics of a connection which disables end-to-end security usage of IPsec and TLS.

4 Security Solutions for BSM PEPs

The following subsections examine the detailed impact of transport, network and link layers security on PEP operations. The issues raised here apply to both distributed and integrated PEP solutions.

A. Interworking between PEPs and transport/application layer security systems

As shown in Figure 4, security can be implemented above the transport layer such as using the Secure Socket Layer (SSL) or its variant call Transport Layer Securiy (TLS) [13]. Also application layer security can be applied such as secure web services [14].

Transport/application layer security will work with TCP PEPs (as described in section I) because the TCP header is not encrypted by the security system. As such, the TCP PEP will function properly and seamlessly. However, if HTTP acceleration is used (application layer PEPs), then there is a problem regarding interworking with security. The reason is that application layer data will be encrypted by the security system. Hence, it will not be possible to perform techniques such as HTTP prefetching, caching and header and payload compressions (described in section I).

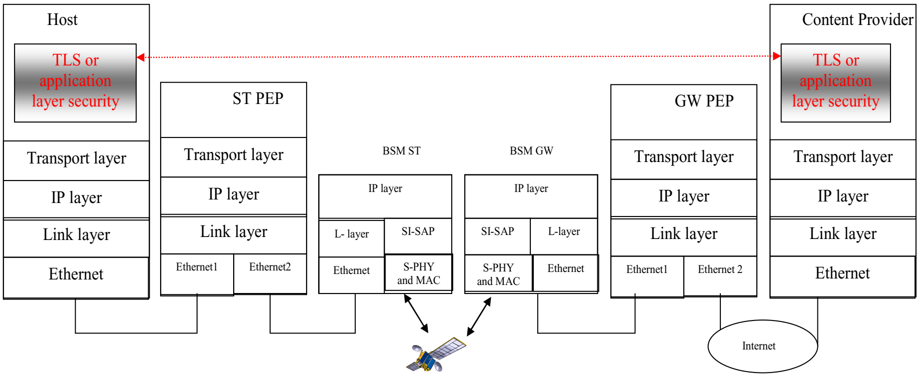

Fig. 4, 5. Distributed PEP implementation with transport/application layer security

B. Interworking between PEPs and IPsec

End-to-end network layer security (such as IPsec) will encrypt the TCP header and user data. Therefore TCP PEPs will not be able to perform techniques such as TCP spoofing, ACK reduction and flow control (described in section I). In addition, the HTTP acceleration will not be able to perform HTTP prefetching, caching and compression. The reason is the encryption of IP packets via IPsec's ESP header (in either transport or tunnel mode) renders the TCP header and payload unintelligible to the PEPs. Without being able to examine the transport or application headers, the PEP may not function optimally or at all [15]. Thus a user or network administrator must choose between using PEPs or using IPsec.

However there are some steps which can be taken to allow the use of IPsec and PEPs to coexist. If an end user can select the use of IPsec for some traffic and not for other traffic, PEP processing can be applied to the traffic sent without IPsec. Another alternative is to implement IPsec over the satellite link between the two PEPs of a distributed PEP implementation (Figure 6). This is not end-to-end use of the IPsec, but it will protect the traffic between the two PEPs.

As shown in Figure 6, PEPs can be used successfully with IPsec in tunnel mode between the BSM ST/Gateway. Here the encryption is performed on incoming traffic after the PEP operations and decryption is performed on outgoing traffic before the PEP operations. In terms of overheads, IPsec tunnel mode requires an extra IP header, where basic IPv4 header is 20 bytes and IPv6 header is 40 bytes.

Also IP multicasting over satellites can exploit the broadcast nature of satellites. However, secure multicasting with IPsec (in tunnel mode) has two more added implications: First, IP multicast becomes effectively point-to-point connections between the the IPsec tunnel ends; second manual keying only is used. Therefore, the recently published RFC 5374 (multicast extension to IPsec) provides an optional extension to IPsec to resolve these issues. However, the multicast extensions to IPsec might not be available on all BSM ST, Gateway or router equipment.

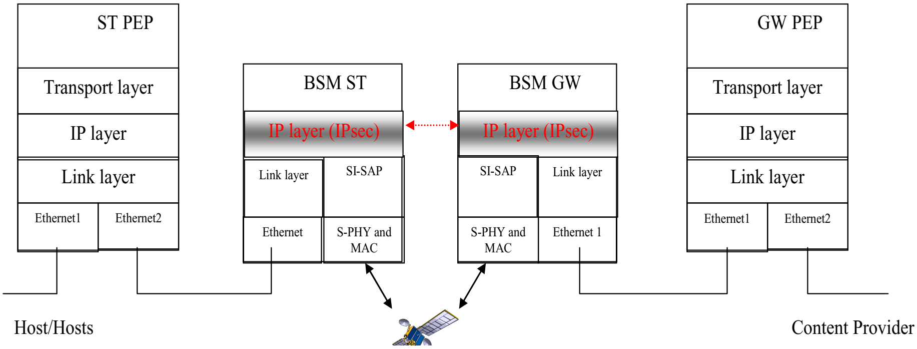

Fig. 6. Distributed PEP implementation with satellite link IPsec security

C. Interworking between PEPs and link layer security systems

As shown in Figure 7, link layer security mechanism can be used such as DVB-RCS [1] security or Unidirectional Link Encapsulation (ULE) security [16]. Here TCP and application layer PEPs will work seamlessly over the secure satellite link. The reason is TCP header and user data are handled in clear text (no encryption) in the Gateway PEP. Then, the satellite link layer security is only applied between the BSM ST and GW (satellite terminals). Finally, the TCP header and user data are handled in clear text (no encryption) in the ST PEP.

Although link layer security does not provide the desired end-to-end security, it is more efficient than using IPsec (in tunnel mode). It also can provide extra security functions that are not possible IPsec or upper layer security such user identity hiding (such as IP and MAC addresses). This allows providing strong privacy service over the satellite broadcasting link. Further details on BSM link layer security is presented in the next section.

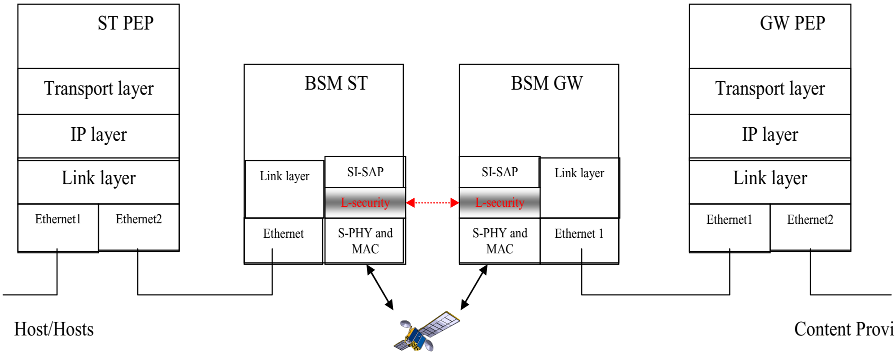

Fig. 7. Distributed PEP implementation with satellite link layer security

5 BSM Link Layer Security Architecture Suitable for PEPs

As shown in Figure 8 [15], the ST and Gateway PEPs can operate seamlessly with link layer security (below SI-SAP) such as DVB-RCS with Multi Protocol Encapsulation (MPE) or Unidirectional Lightweight Encapsulation (ULE) RFC 4326. In addition, Figure 8 provided detailed key management architecture and security interactions across the BSM SI-SAP interface. The data encryption (and data integrity check) is performed below the SI-SAP, while the key management is performed above the SI-SAP within entities co-located with the BSM ST and Gateway (satellite terminals).

Also Figure 8 show the client (user) authentication process (supplicant, authenticator and Authentication server entities), where secure link layer is used to carry authentication information (such as user name and password) between supplicant and authentication server. This authentication is independent of the PEPs operations.

The SI-U-SAP (User) interface is used to communicate secure user information (this includes TCP headers and user data). Also the client authentication messages use the SI-U-SAP interface. However, the key management information is passed through the SI-C-SAP (Control) interface, because this function is normally performed during connection establishment phase.

Both authentication server and the BSM Network manager communicate with the BSM Network Control Centre (NCC) regarding security and authorization. These

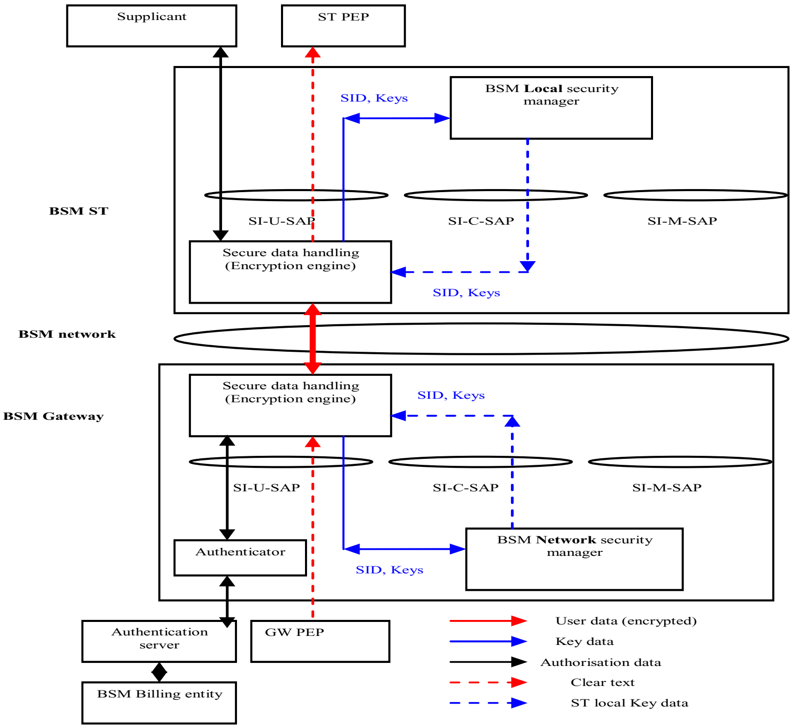

Fig. 8. Mixed link layer BSM security entities

interactions are not shown here in order to simplify the diagram. The Security association identity (SID) must be used in all security management message exchanges.

Thus link layer security can work seamless with TCP and application layer PEPs and provide strong access control to the satellite network resources.

6 Conclusion

The ETSI BSM standardisation work is focussed on the efficient transport of IP data streams and on how to interoperate resulting satellite networks with terrestrial IP networks. The paper presented the current work in ETSI BSM group in defining the PEP architecture for BSM networks. The ST/Gateway PEP protocol stack has been shown together to scenarios for distributed and integrated PEPs. Also an overview of TCP and HTTP PEP techniques and past work related to interworking between PEPs and security system are presented.

Detailed PEP and BSM terminal configurations with transport layer security (TLS), network layer (IPsec) and satellite link layer (DVB-RCS or ULE) are presented. The strengths and weakness of each solution has been analysed. The main findings of this paper is that application layer PEPs (such as HTTP accelerators) will not work with end-to-end TLS or IPsec. It will only work with link layer security.

The TCP PEP will not work with IPsec (end-to-end configuration), but it will work with TLS and link layer security.

Therefore link layer security is the most suitable solution for PEPs, where additional identity hiding (IP addresses and satellite terminal MAC addresses) can be provided. The paper presents a detailed BSM security architecture for link layer security below the BSM SI-SAP interface and key management above the SI-SAP interface.

Acknowledgement

This work was sponsored by the European Telecommunications Standards Institute (ETSI) [17] and supported by the EU Information Society Technologies SATNEX project [18].

References

[1] ETSI. Digital Video Broadcasting (DVB); DVB specification for data broadcasting. ETSI EN 301 790 V1.4.1. Interaction channel for satellite distribution systems (April 2005)

[2] IETF document: Performance Enhancing Proxies Intended to Mitigate Link-Related Degradations, RFC 3135, http://www.ietf.org

[3] http://www.xiplink.com/IMG/pdf/XipLink_Internet_over_Satellite_Optimization-R2.pdf

[4] http://www.spacebel.be/FR/Space/FastSatDataSheet.pdf

[5] http://www.direcway.com/HUGHES/Doc/0/SIKPBJS69O6KP42VCE4K4ER2BF/Hughes%20PEP_H35661_A4_LR_091206.pdf

[6] ETSI TS 102 292, Broadband Satellite Multimedia (BSM); Functional Architecture

[7] Caini, C., et al.: PEPsal: A Performance Enhancing Proxy for TCP Satellite Connections. IEEE A&E Systems Magazine (August 2007)

[8] I-PEP specifications, Issue 1a. Satlabs group recommendations (October 2005), http://www.satlabs.org

[9] ETSI TS 102 463: Broadband Satellite Multimedia (BSM); Interworking with IntServ QoS

[10] ETSI TS 102 464: Broadband Satellite Multimedia (BSM); Interworking with DiffServ QoS

[11] 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)

[12] Bellovin, S.: Probable plaintext cryptanalysis of the IPSecurity protocols. In: Proceedings of the Symposium on Network and Distributed System Security (February 1997)

[13] Dierks, T., et al.: The TLS Protocol Version 1.2, RFC 5246 (August 2008)

[14] Moser, L., et al.: Building Dependable and Secure Web Services. Journal of Software 2(1) (February 2007)

[15] ETSI TS 102 465, Broadband Satellite Multimedia (BSM); Security Functional Architecture

[16] Cruickshank, H., Pillai, P., Noisternig, M.: Security requirements for the Unidirectional Lightweight Encapsulation (ULE) protocol, Internet Draft (draft-ipdvb-sec-req-09.txt) (August 2008)

[17] ETSI home page: http://portal.etsi.org/Portal_Common/home.asp

[18] Satnex project home page: http://www.satnex.de