Network Layer Security: Design for A Cross Layer Architecture

S. Ramachandran[±], G. Fairhurst[±], M. Luglio[*], C. Roseti[*], S. Provenzano[*]
[±]*Electronic Research Group, University of Aberdeen, Scotland, AB24 3UE*
[*] *Electronics Engineering Department, University of Rome "Tor Vergata", Italy*
{raj, gorry}@erg.abdn.ac.uk, luglio@uniroma2.it, roseti@ing.uniroma2.it

***Abstract*- Traditional modular layering schemes have served a major part in the development of a variety of protocols. However, as the physical layer impairments become more unpredictable, a cross layer design (CLD) which is dynamic in nature provides better performance. CLD introduces new challenges in protocol design as well as in the area of security.**

Using numerical analysis, we show that a link layer design employing header compression and cross layer signalling to protect protocol headers can limit packet discarding. This paper also reviews the IPsec protocol and describes how IPsec can be modified for cross layer architecture.

Key words: Cross layer, UDP-Lite, IPsec, CL-IPsec, Header protection

I. INTRODUCTION

Traditionally network systems design has followed the Open Systems Interface (OSI) model. In this model the complex task of host-to-host networking is divided into different logical layers, and information is passed between adjacent logical layers through a specific interface (service access point). Today a variety of communication mediums (wired and/or wireless) are used to relay information. This heterogeneity in the network infrastructure may cause information to be lost due to either erratic channel behaviour (e.g. scintillation errors, signal fade etc.) or a processing glitch in the intermediate systems. To cope with such dynamic behaviours, next generation network systems design needs a reference model that is more flexible. One such model is the Cross Layer Architecture (CLA).

CLA in a "nut-shell" can be defined as a design approach where, non-adjacent layers of an OSI reference model co-ordinate in order to optimize system performance. This design approach contradicts the OSI reference model, where the protocols in different layers function independent to each other and only adjacent layers can communicate with one another through well known interfaces. In a CLA, it is assumed that the layer(s) can tolerate errors to a certain magnitude in parts of its payload. Modern multimedia codecs (e.g. AMR [1], H.264 [2]) are designed to be error resilient. Other applications such as reliable multicast transport can use various error/erasure correction codes to protect against channel impairments.

Today UDP is the preferred transport protocol to deliver multimedia as well as multicast packets over the Internet. However, due to its stringent error check, even single bit error may lead to packet loss. A transport protocol called UDP-Lite, that uses partial checksum, was therefore designed to alleviate this inherent problem of UDP. UDP-Lite inspects error on only part of the packet identified as sensitive to errors by the checksum coverage field, and ignores errors in the remaining parts of the packet. However to take advantage of UDP-Lite, modifications are required at the lower layers to allow corrupted packets to be delivered to the higher layers. Security is paramount in today's Internet. A security architecture that is compliant with UDP-Lite needs to be considered.

The structure of this paper is as follows: the next section introduces the difference between UDP and UDP-Lite. This section also explains the various link layer modifications that are required when using UDP-Lite. Section 3 describes a security architecture using IPsec that is compliant with a CLA approach using UDP-Lite. This section also addresses the use of header compression with IPsec. Conclusion and future work is explained in Section 4.

II. UDP Vs. UDP-Lite

Due to its low protocol overhead (8 bytes) and processing overhead User Datagram Protocol [3] has found its usage in various delay sensitive as well as streaming application. Many of these applications can tolerate bit errors in the data payload better than the loss of a full packet. For instance modern audio/video codec such as Reversible Variable Length Codes (RVLC) [4] can extract useful information from blocks of corrupt data to conceal the effect of error. This can yield a better degree of visual or audio experience. Other examples are reliable multicast protocols which can employ packet-level forward error correction (FEC) codes to reliably recover from errors and/or erasures. However due to the strict error check provided by UDP the entire packet will be dropped in case of bit errors. To solve this problem the IETF standardized a protocol called UDP-Lite [5]. As shown in Figure. 1 the difference between UDP and UDP-Lite is that the 16-bit Length field in the UDP is replaced by a 16-bit Checksum Coverage field.

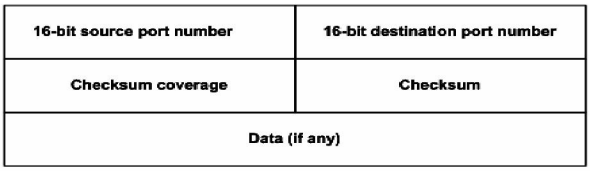

Figure. 1. UDP-Lite header

When using UDP-Lite, a packet is divided into sensitive and insensitive parts. An application uses the Checksum Coverage field, to indicate the number of bytes from the start of the UDP-Lite header that are to be considered sensitive to bit errors. Since the receiver only calculates checksum over the sensitive part any bit errors in the insensitive portion of the packet is ignored. The minimum coverage length is 8 bytes, which only includes the UDP-Lite header whereas a coverage equal to zero indicates that the checksum covers the entire packet [5].

A. Limit packet discarding using UDP-Lite

Work described in [6] has shown the advantage of using UDP-Lite. However to achieve this it is important that the sensitive bytes are delivered error-free. [7] has shown that, it is common for a packet header to be corrupt. For reliable delivery of sensitive bytes, following techniques can be used:

Header compression and Partial checksum:- When using UDP-Lite it is essential that the lower layers do not drop the packet due to errors in the insensitive part. A partial error detection scheme, as shown in Figure. 2, is therefore required at the lower layers (e.g. link layer). Implicit cross layer signalling techniques can be used to modify link layer to provide partial error check.

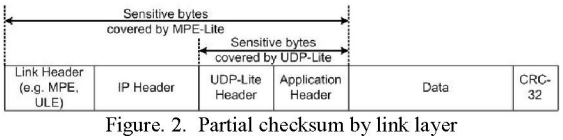

Figure. 2. Partial checksum by link layer

Works described in [8-10]have shown that using a partial checksum scheme, which detects error in the link layer header and the sensitive part of the frame improves the probability of packet delivery to the higher layers.

In some applications the overhead due to protocol headers can be larger than the application data itself (e.g. VoIP). Such an over head can be reduced by use of compression algorithms such as RoHC, IPHC, etc. Performing HC over sensitive bytes not only reduces the channel utilization, but it also reduces the probability of errors in sensitive bytes.

Header compression with partial checksum and Header Protection: - In case of networks where error patterns vary rapidly with time (e.g. mobile satellite nodes), the sensitive bytes can still be in error [7]. One way of protecting the sensitive bytes is by using a strong forward error correction (FEC) code. The work described in [11] has shown in detail how such a scheme can be useful for both error-tolerant applications as well as for bulk data transfer. The model described here uses a combination of header compression, partial checksum and header protection as illustrated in Figure 3.

A sample architecture that uses this technique is given in [12], where Robust Header Compression (RoHC) [13] was used to compress the protocol headers (RTP/UDP/IPv6) and Joint Source Channel coding and decoding (JSCC/D)[14] was used to provide the necessary error protection of the sensitive bytes at the physical layer.

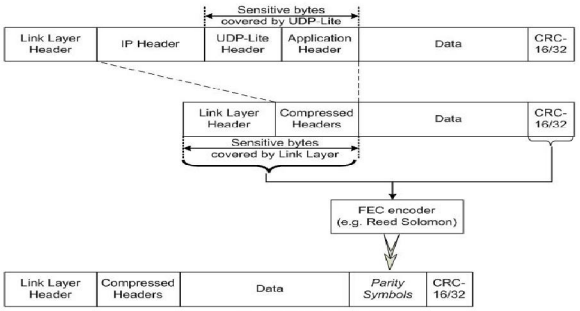

Figure 3. MPHP with HC

Other mechanisms, as described in [15] have been proposed, where the a packet is divided into different frames, based on the sensitivity information from the link layer, and a higher coding and modulation is used for the sensitive parts w.r.t. insensitive parts. A process to de-multiplex these frames needs to be designed.

B. Packet loss analysis for various schemes

To evaluate the performance of the above schemes, we compute the packet loss ratio at the transport layer as a result of the varying link layer bit error rates (BER). For the purpose of analysis, we have considered the use of a reliable multicast protocol called FLUTE [16] over UDP-Lite and the link layer protocol considered is the Unidirectional Lightweight Encapsulation (ULE)[17].

The coverage length at the link layer include the ULE header (4 bytes), IP header (20 bytes), UDP-Lite header (8 bytes), FLUTE header (including extension headers, 44 bytes) and the link checksum (4 bytes), which is a total of 80 bytes. In the schemes using header compression the coverage length can be reduced to 40 bytes, i.e. compressing the IP/UDP-Lite header to 4 bytes [6], the FLUTE general header and the extension headers can be compressed to 28 bytes based on the methods described in [18]. The remaining 8 bytes corresponds to the uncompressed link header and CRC-32.

The link layer schemes discussed in the previous section, results in transport layer observing both erasures and errors.

Assuming uniform error distribution the packet loss rate at the transport layer can be described using the following equation

$$PLR_{TL} = 1-(1-BER_{link})^{(CL_{link})} + BER_{link}\times(1-BER_{link})^{(CL_{link})} \quad (1)$$

where,

PLR_{TL} = packet loss ratio at transport layer

BER_{link} = bit error ratio at link layer

CL_{link} = no: of bits covered by link CRC

When header protection is not used, the BER_{link} is the residual BER after the demodulation and/or decoding at the physical layer. On the other hand with header protection, the BER_{link} is the decoder error probability of an FEC code at the link layer. Here we assume the use of a Reed Solomon code, whose upper bound decode error probability is given by

$$BER_{link} = \sum_{i=n-k+1}^{n} \binom{n}{i} \cdot BER_{phy}{}^{i} \cdot (1-BER_{phy})^{n-i} \quad (2)$$

where,

BER_{phy} = bit error ratio after demodulation and decoding at physical layer

n = total encoded symbols

k = original source symbols (header bytes to protect)

Figure. 4. shows the packet loss ratios at the transport layer using various link layer schemes. Two observations were made from this analysis. Firstly, although header compression improves the probability of packet delivery when compared to the scheme without header compression, e.g. approx. 42% for BER 10^{-3}, this gain margin decrease as the link layer BER increases.

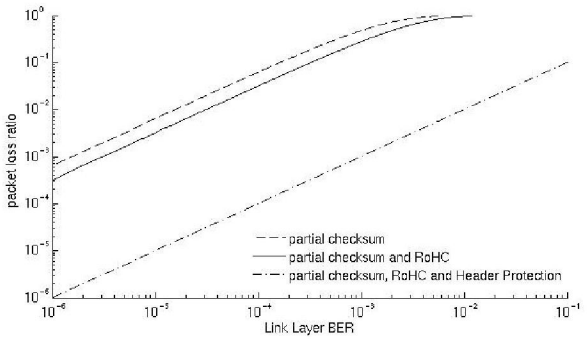

Figure. 4. Bit errors in transport layer for different schemes

Secondly, with a code rate of 0.3, i.e. an additional overhead of 80 bytes in the form of parity symbols, the scheme with header compression and protection outperformed the other schemes by orders of magnitude. The use of header compression and protection not only reduces the protocol header overhead, but the additional protection ensures that the errors in the compressed header are corrected. An error-erasure correction code (e.g. [19]) can be used by FLUTE to correct residual bit errors in its payload.

III. Network Security

In the framework of Internet security, IETF has standardized the IP security protocol (IPsec) [20] with the aim to offer inter-operable cryptographically-based security services (confidentiality, authentication, integrity and non-repudiation) while continuing to use the existing infrastructures.

Such services are provided through an authentication protocol, named Authentication Header (AH) [21] a confidential protocol, named Encapsulating Security Protocol (ESP) [22] and an Internet Security Association Establishment and Key Management Protocol (ISAKMP) [23]. These protocols have been designed as an IPv4 upgrade and as predefined security for IPv6.

The used cryptographic/authentication algorithm and keys of the IPsec services are defined through Security Associations (SAs). A single SA can support the use of AH or ESP, but not both. IPsec operates in two modes: transport and tunnel mode. The former is used between end-systems and adds a new header (AH or ESP) to the IP guaranteeing the protection of the IP payload. In tunnel mode, on the other hand, the end-system delegates the security service to the gateway. In this mode, AH or ESP header encapsulates the entire IP packet and a new IP encapsulation is formed, whose destination and source addresses can be different from those of the encompassing IP packet.

AH jointly provides authentication and integrity by adding to the protected datagram an additional block, called "Integrity Check Value" (ICV), which can be either a Message Authentication Code (MAC) or a digital signature. AH format presents the following fields:

- *Next Header* (1 byte) It defines the type of the payload that follows immediately the AH header (i.e., UDP, TCP);
- *Payload length* (1 byte) It indicates the length of the AH payload;
- *Reserved* (2 bytes) This field is reserved for future use;
- *SPI field* (4 bytes) The Security Parameter Index field is used to identify the appropriate SA;
- *Sequence Number* (4 bytes) Sequence Number used for anti-replay;
- *Authentication Data* (variable) Authentication data using at least HMAC-MD5 and HMAC-SHA1.

ESP ensures the confidentiality service, by adding to the field used in AH, the following fields:

- *Initialization Vector* Vector used by the ESP encryption algorithms.
- *Padding* Padding bits are used to align the payload and the payload and the two following fields on a 32 bit boundary, as requested by the encryption algorithm.
- *Padding length* It indicates the size of the used padding (in bytes).

B. Cross-Layer IPsec for UDP-Lite

Traditional IPsec authenticates (and optionally encrypts) the entire IP payload. This means that corruption of any part of the IP payload causes authentication failure and results in packet drop. In other words, IPsec assumes that the entire IP payload is sensitive to unauthorized bit changes (due to either bit errors or malicious attacks). This conflicts with UDP-Lite behaviour which can tolerate bit errors in its payload.

The proposed Cross Layer IPsec (CL-IPsec) aims to adapt IPsec for UDP-Lite based applications. The behaviour of CL-IPsec is dependent on the cross layer signalling between network layer and higher layers. Specifically, IPsec needs to receive both explicit signalling from application, indicating the use of UDP-Lite, and implicit signalling from transport layer to get the coverage length value and then perform the security operations accordingly.

Considering the AH protocol in transport mode, and based on the aforementioned signalling scheme, a CL-IPsec scheme of implementation is shown in Figure. 5. where the insensitive part only involves the RMT payload. It allows partial authentication involving only AH, UDP-Lite and other sensitive bytes. To achieve this, the input of the ICV algorithm should be modified in order to consider only the following fields: new IP header (if IPsec is running in tunnel mode), AH header, IP header (excluded the mutable fields: Flags, Fragment Offset, Time to Live and Header Checksum [21]) and the sensitive part of the UDP-Lite packet. In this way, even though bits belonging to the insensitive part are corrupted, IPsec forwards the packet to the higher layers. CL-IPsec adaptation allows accessing the checksum coverage field within the UDP-Lite header through an implicit cross-layer interaction with the transport layer. Such an interaction is possible because the position of the checksum coverage is fixed within the UDP-Lite header and a priori knowledge of the AH header size. Note that IPsec is in general not able to distinguish IP header and IP payload.

Once checksum coverage is made evident to AH, it is possible to change the input of ICV algorithm accordingly. In case the ESP protocol is used, without exploiting its cryptographic service, the CL-IPsec approach requires modifications in order to take into account the presence of the ESP trailer. On the other hand, if confidentiality is required, the distinction of a sensitive and an insensitive part does not make sense.

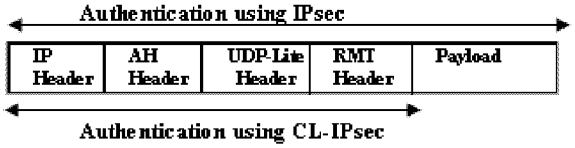

Figure. 5. Partial Authentication using CL-IPsec in Transport mode

B. Header compression for IPsec

IPsec provides various security services at the cost of increased overhead. Especially in the tunnel model, the IPsec overhead implies inefficient bandwidth utilization [24, 25]. This drawback can be mitigated by using Header Compression (HC) protocols. Header compression over IPsec (HCoIPsec) [24] aims to reduce overhead, without compromising the security services provided by IPsec. HCoIPsec framework relies on two assumptions:

1. Existing HC protocols are considered;

2. HC protocols operate at the IPsec SA endpoints (HC applied in a SA basis).

Since existing HC protocols compress packets on a hop-by-hop basis, HCoIPsec requires the extension of the HC functionalities in order to operate at IPsec SA endpoints. Furthermore, HCoIPsec framework proposes that the configuration of the HC parameters is accomplished by the SA management protocol (i.e. IKEv2 [23] while compressed packet can be identified through the Next Header field of the security protocol (AH or ESP).

Performing HCoIPsec, outbound IP traffic is first appropriately compressed and then encrypted/authenticated. Similarly, inbound IP traffic is first decrypted/authenticated and then decompressed [24]. An example concerning AH in tunnel mode is shown in Figure. 6.

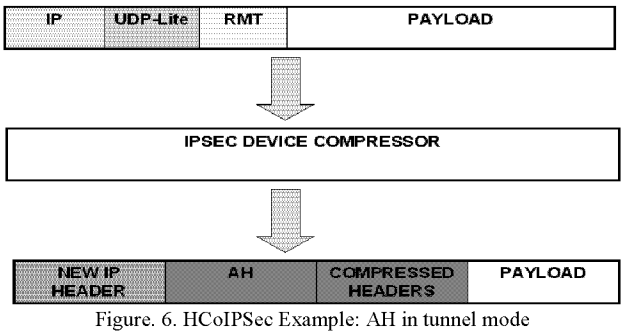

Figure. 6. HCoIPSec Example: AH in tunnel mode

IV. Conclusion & Future Work

In the first half of this paper we have focused on the different link layer designs that are applicable for multimedia as well as multicast applications using cross layer architecture. The impact of different link layer designs on the packet loss ratio at the transport layer was studied. The results have shown that the scheme using a combination of header compression and protection with partial link checksum outperforms other schemes by orders of magnitude. This conclusion achieved using numerical model is also coherent with the findings in [12, 26]

The second half of this paper identified the drawback of using IPsec for applications designed for UDP-Lite protocols. To alleviate the problem, we have proposed a cross layer signalling scheme which ensures that only the IP header and the sensitive bytes identified by the coverage length of the UDP-Lite is authenticated by the ICV algorithm. This scheme, however, works with only AH or with ESP with authentication and no encryption. When ESP uses encryption, it implies that all the parts in the IP payload is sensitive to change (due malicious attack or bit errors). Hence such cases are ignored in our proposed cross layer architecture.

As part of the future work, we will investigate the impact of various link layer schemes on FLUTE protocol. Initially FLUTE will be modified to use error and erasure correction codes, following this a header compression profile, either based on [18] or other techniques will be considered.

The IPsec implementation in Linux Kernel 2.6.20 will be modified to the CL-IPsec protocol. Furthermore based on a header compression algorithm, CL-IPsec will also be adapted.

Acknowledgement

This work is part of a joint collaboration between the University of Aberdeen and University of Rome "Tor Vergata". This work is funded as part of the JA2240 work package of the European IST-FP6 project: "SatNEx II – Satellite Communications Network of Excellence II".

References

[1] 3GPP TS 26.091:, "Mandatory speech codec speech processing functions; AMR speech codec; error concealment of lost frames," Tech. Rep. 26.091 V6.0.0, 2004.

[2] D. Marpe, T. Wiegand and G.J. Sullivan, "The H.264/MPEG4 Advanced Video Coding Standard and its Applications," *IEEE Communications Magazine,* vol. 44, pp. 134-143, August 2006.

[3] J. Postel, "User datagram protocol," IETF, Tech. Rep. RFC 768, August, 1980.

[4] J. Wen and J.D. Villasenor, "Reversible variable length codes for efficient and robust image and video coding," in *In Proceedings of IEEE Data Compression Conference,* Apr 1998, pp. p.g. 471-480.

[5] L-A. Larzon, M. Degermark, S. Pink, L-E. Jonsson and G. Fairhurst, "The lightweight user datagram protocol (UDP-lite)," IETF, Tech. Rep. RFC 3828, July, 2004.

[6] W. Stanislaus, G. Fairhurst and J. Radzik, "Cross layer techniques for flexible transport protocol using UDP-lite over a satellite network," in 2005, pp. 706- 710.

[7] M. Rossi, "Evaluating TCP with Corruption Notification in an IEEE 802.11 Wireless LAN," M.S Thesis, Institute of Computer Science, University of Innsbruck. November 2006.

[8] L-A. Larzon, M. Degermark and S. Pink, "UDP lite for real time multimedia applications," in June 1999,

[9] A. Servetti and J.C. De Martin, "Error tolerant MAC extension for speech communications over 802.11 WLANs," in *In Proceedings of IEEE 61st Semiannual Vehicular Technology Conference (VTC),* May 2005, pp. p.g. 2330-2334.

[10] E. Masala, M. Bottero and J.C. De Martin, "Link-level partial checksum for real-time video transmission over 802.11 wireless networks," in *In Proceedings of 14th International Packet Video Workshop (PVW),* Dec 2004,

[11] F. Arnal, L. Dairaine, J. Lacan and G. Maral, "Cross-layer reliability management for multicast over satellite," *Computer Networks and ISDN Systems,* vol. No. 48, pp. p.g. 29-43, May 2005.

[12] M.G. Martini, M. Mazzotti, C. Lamy-Bergot, J. Huusko and P. Amon, "Content adaptive network aware joint optimization of wireless video transmission," *Communications Magazine, IEEE,* vol. Vol. 45, pp. p.g. 84-90, Jan. 2007.

[13] C. Bormann, C. Burmeister, M. Degermark, H. Fukushima, H. Hannu, L-E. Jonsson, R. Hakenberg, T. Koren, K. Le, Z. Liu, A. Martensson, A. Miyazaki, K. Svanbro, T. Wiebke, T. Yoshimura and H. Zheng, "RObust header compression (ROHC): Framework and four profiles: RTP, UDP, ESP, and uncompressed," IETF, Tech. Rep. RFC 3095, July 2001.

[14] J.L. Massey, "Joint source and channel coding," in *Communication Systems and Random Process Theory, NATO Advanced Studies Institutes Series E25* J. K. Skwirzynski editor, Ed. Sijtho & Noordho , Alphen aan den Rijn, The Netherlands: 1978, pp. p.g. 279-293.

[15] G. Fairhurst, M. Berioli and G. Renker, "Cross-layer control of adaptive coding and modulation for satellite Internet multimedia," *International Journal of Satellite Communications and Networking,* vol. 24, pp. 471-491, 2006.

[16] T. Paila, M. Luby, R. Lehtonen, V. Roca and R. Walsh, "FLUTE - file delivery over unidirectional transport," IETF, Tech. Rep. RFC 3926, October, 2004.

[17] G. Fairhurst and B. Collini-Nocker, "Unidirectional lightweight encapsulation (ULE) for transmission of IP datagrams over an MPEG-2 transport stream (TS)," IETF, Tech. Rep. RFC 4326, December 2005.

[18] R. Walsh, J-P. Luoma and A. Saaranen, "Method and System for header compression," US. US 2005/0160184 A1, July 21, 2005.

[19] R.M. Zaragoza, "http://www.eccpage.com/," Aug 21, 2006. 2006.

[20] K. Seo and S. Kent. (Dec. 2005, Security architecture for the internet protocol. IETF,

[21] S. Kent, "IP authentication header," IETF, Tech. Rep. RFC 4302, December 2005.

[22] S. Kent, "IP encapsulating security payload," IETF, Tech. Rep. RFC 4303, December 2005.

[23] C. Kaufman, "Internet key exchange (IKEv2.0)," IETF, Tech. Rep. RFC 4306, December 2005.

[24] E. Ertekin, C. Christou, R. Jasani and B.A. Hamilton, "Integration of header compression over IPsec security associations," IETF, Tech. Rep. Internet-Draft, draft-ietf-rohc-hcoipsec-04, February 2006.

[25] M. Luglio and C. Roseti, "Network security and performance evaluation of ML-IPSec over satellite networks," in *In Proceedings 12th Ka and Broadband Comunications Conference,* Sep 2006,

[26] F. Arnal, L. Dairaine, J. Lacan and G. Maral, "Multi-protocol header protection (MPHP), a way to support error-resilient multimedia coding in wireless networks," in *High Speed Networks and Multimedia Communications, 7th IEEE International Conference, HSNMC 2004,* July 2 2004, pp. 740-749.