A Cross-Layer Architecture for Satellite Network Security: CL-IPsec

C. Roseti•, M. Luglio•, S. Provenzano•, S. Ramachandran±, and G. Fairhurst±

Abstract— **Cross-layer architectures (CLAs) are proposed to improve performance in networks where physical layer impairments are unpredictable and provision of security services may be challenging, as in satellite networks.**

This paper proposes an extension to the IPsec protocol, named Cross-Layer IPsec (CL-IPsec), able to provide authentication and integrity services through a cross-layer architecture when the adopted protocol is UDP-Lite. This is suitable for multicast applications that are cost-effectively provided by satellite systems.

A satellite emulation platform has been used to validate the CL-IPsec implementation and to evaluate the performance improvement derived from the proposed CLA.

Index Terms—**Cross-Layer, IPsec, security and UDP-Lite**

I. Introduction

The Internet reference model is based on a layering paradigm: each layer is independently optimized and exploits services provided by lower layers without information on how such services are derived. Implementing this model in wireless networks (i.e. satellite, WiFi, WiMAX, 4G, etc.), such a layer independency leads to non-optimal performance due to the dynamically-varying channel characteristics. As a consequence, data packets may be delivered with corrupted bits and are usually discarded by the link. This implies a degradation of the performance perceived at upper layers, although many audio/video applications use an encoding format able to handle single-bit errors in the data payload better than the loss of a full packet. To efficiently cope with both dynamic behaviors and the particular needs of the applications, next generation systems requires a more flexible architecture, such as the Cross-Layer Architecture (CLA) [1][2]. CLA is a design approach allowing non-adjacent layers of an OSI reference model to cooperate to optimize system performance.

A transport protocol called UDP-Lite [3] was designed to replace UDP [4] in a CLA where applications can tolerate errors up to a certain amount in their payloads. UDP-Lite allows division of a datagram into a sensitive and an insensitive part. An application can use the Checksum Coverage field (replacing the length field in the UDP header) to indicate the number of bytes from the beginning of the UDP-Lite header that must be considered sensitive to bit errors. Receivers calculate the checksum only over the sensitive part, then datagrams with only bit errors in the insensitive part are forwarded to the application.

When using UDP-Lite, it is essential that lower layers do not drop a datagram with bit errors in the insensitive part. In other words, lower layers (e.g. link layer) must be aware of the UDP-Lite partial checksum [5][6][7][8]. An example of such a link layer frame type is the HDLC UIH [9] frame type, currently supported by the 3GPP architecture. Here the CRC covers a pre-agreed minimum number of bytes and any error in the remaining frame is ignored. For reliable delivery of sensitive bytes while tolerating error in the insensitive bytes, the following techniques can be used: header compression and partial checksum, header compression with partial checksum and header protection.

One more challenging aspect for a CLA is the provision of security services [5]. IETF has standardized the IP security protocol (IPsec) suite to offer inter-operable security services while continuing to use the existing infrastructures [10]. Such security services refer to the whole IP packet through an authentication protocol, named Authentication Header (AH) [11], and a confidential protocol, named Encapsulating Security Protocol (ESP) [12]. In practice, IPsec assumes that the entire IP payload is sensitive to bit errors making useless cross-layer adaptation at the other layers.

In this paper, we propose a Cross-Layer IPsec (CL-IPsec) design aiming to adapt IPsec to UDP-Lite based applications in a CLA framework. Specifically, CL-IPsec receives from the transport layer the UDP-Lite coverage length and applies the security services only on the sensitive part of the packet. Throughout this work, CL-IPsec has been implemented for AH protocol in transport mode.

The rest of the paper is organized as follows: Section II presents the reference protocol stack; Section III describes CL-IPsec design; Section IV provides details on CL-IPsec implementation; Section V shows performance results of the proposed CLA over a satellite emulation platform and Section VI provides conclusions and resumes guidelines for the future work.

• Electronics Engineering Department, University of Rome “Tor Vergata”, Italy. Email: luglio@uniroma2.it, {silvio.provenzano, roseti}@ing.uniroma2.it.
±Electronic Research Group, University of Aberdeen, Scotland, AB24 3UE. Email: {gorry,raj}@erg.abdn.ac.uk.

978-1-4244-2204-3/08/$25.00 ©2008 IEEE

II. Reference Protocol Stack

The reference protocol stack herein considered is shown in the Figure 1.

Application layer – At the application layer, error-tolerant protocols are considered.

Modern multimedia codecs are designed to be error resilient. For instance, MPEG video coding [13][14] sends data using three different frame types: I-, P- and B-frames. I-frames hold information about an entire video frame, while P- and B-frames only include the differences to other frames. Usually it is better to deliver damaged P- and B- frames than discarding them. MPEG-4 [14] provides higher compression with greater error robustness at a large range of bit rates. MPEG-4 video standard includes new features such as object-based coding, error resilience and improved compression. Error resilient tools in MPEG-4 video do not reduce errors like FEC or ARQ, but reduce quality degradation caused by errors (i.e. use error concealment).

Reliable Multicast Transport (RMT) protocols [15][16][17] use various error/erasure correction codes to protect against channel impairments. Mass file delivery consists of one-to-many data communication using UDP transport over IP. Using FLUTE [18] defines a specific file delivery application of Asynchronous Layered Coding (ALC) [19], adding the following specifications: definition of a file delivery session built on top of ALC, including transport details and timing constraints, in-band signaling of the transport parameters of the ALC session, in-band signaling of the properties of delivered files and details associated with the multiplexing of multiple files within a session.

Transport layer – UDP-Lite is used instead of UDP. The main difference between UDP and UDP-Lite is the partition of each packet into two parts: sensitive and insensitive. Errors in the sensitive part result in a packet drop, while packets with errors in the insensitive part are forwarded to the application. Further details are provided in Section III.

Network layer – Either IPv4 or IPv6 can be alternatively selected. IP checksum is computed only on the IP header field to verify that the IP header was not damaged. In IPv4, such a checksum is mandatory, while IPv6 relies on both the link CRC and transport checksum to assure IP header integrity.

In addition, CL-IPsec is used to provide security services at the network layer. Herein, AH protocol in the transport mode is implemented. As a benchmark standard IPsec AH protocol is considered.

Link layer – Link layer must calculate the CRC according to the payload type. That is, in case of UDP-Lite a *partial checksum* must concern only the link layer header and UDP-Lite sensate part. *Header protection* techniques can help to avoid bit errors in the sensitive part (possibly used in combination with link header compression)[5].

This paper focuses on the definition of the CL-IPsec AH protocol by proposing also the design of the CL signaling needed to get UDP-Lite checksum value in both sender and receiver sides.

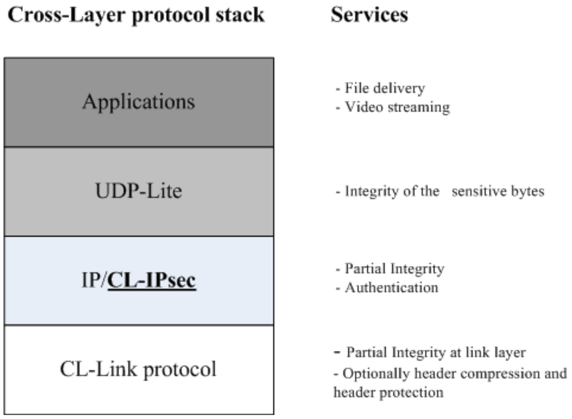
Cross-Layer protocol stack
Services
Applications
UDP-Lite
IP/CL-IPsec
CL-Link protocol
- File delivery
- Video streaming
- Integrity of the sensitive bytes
- Partial Integrity
- Authentication
- Partial Integrity at link layer
- Optionally header compression and header protection

Figure 1: CL protocol stack and related services

III. CL-IPsec design

UDP-Lite and the IPsec protocol suite are intrinsically incompatible. In practice, IPsec performs its security tasks on the entire IP payload, irrespective of the UDP-Lite protocol that identifies a sensitive and an insensitive part [3]. Therefore, IPsec discards all packets with one or more bits corrupted. The UDP-Lite capability to manage corrupted bits in the insensitive part is then prevented resulting in the net performance of UDP-Lite to be similar to UDP.

To avoid the intrinsic incompatibility between IPsec and the partial payload coverage of UDP-Lite, IPsec must identify the UDP-Lite sensitive part within the IP packet (only for UDP-Lite packets), which requires access to the *checksum coverage* field in the UDP-Lite header. This task can be performed through a Cross-Layer interaction between IPsec and UDP-Lite. Furthermore, the integrity check must be performed in accordance with UDP-Lite *checksum coverage*: the UDP-Lite insensitive part should be excluded from the Integrity Check Value (ICV) calculation.

CL-IPsec design is divided into two different phases:

1. Identification of an appropriated cross-layer method;
2. Design of a new algorithm for the ICV calculation.

A. Design of the cross-layer signaling

The first step is to identify the checksum coverage value in the UDP-Lite header.

Cross-Layer methods are mainly classified on the basis of the presence or absence of signaling between the involved protocol layers [2]. Specifically, in an *implicit* cross-layer design, cross-layer interactions are defined in the design phase without any exchange of control information during protocol operations. On the contrary *explicit* cross-layer requires exchange of control information between participating protocol layers. For instance, application explicitly informs transport layer on which bytes are to be considered sensitive; implicit CL signaling can be used to modify link layer to provide partial error check.

From the analysis of IP, IPsec (AH in transport mode) and the UDP-Lite protocol, two particular aspects can be highlighted:

- UDP-Lite header size is fixed (8 bytes);
- IP and AH header sizes are not fixed, however they are known to IPsec.

On this basis, the AH protocol can know the sensitive bytes through the definition of a new “cross-layer” pointer that is dynamically associated to the *Coverage* field in the UDP-Lite header. This makes effective an implicit cross-layer interaction between transport and network layer. Such a modified IPsec (AH) protocol is called CL-IPsec.

B. A new algorithm for the ICV calculation

Once cross-layer signaling is defined, the algorithm for the Integrity Check Value (ICV) calculation must be modified to take into account only sensitive bytes. In particular:

- The insensitive part of the UDP-Lite packet must be excluded from ICV calculation;
- The new ICV algorithm must run only for UDP-Lite packets. Standard procedures are considered for all the other packets (i.e. UDP, TCP) making, CL-IPsec compatible with any other transport protocol.

The routers can modify certain fields in the IP header. These fields are called mutable fields, for example *Type of Service*, *Flags*, *Fragment Offset*, *Time to Live*, *Header Checksum, Explicit Congestion Notification*. All mutable fields are set to zero before computing the ICV at both sender and receiver ends.

In CL-IPsec, this approach is also used to manage bits comprising the insensitive part. Upon receiving UDP-Lite packets, CL-IPsec acquires the checksum coverage value from the UDP-Lite header (implicit cross-layer), and accordingly sets to zero all the bits of the insensitive part before computing the ICV at both sender and receiver sides. In this way, packets with corrupted bits in the insensitive part are not discarded.

C. Security services

IPsec offers a flexible set of security services. These services are:

Data origin authentication/Connectionless data integrity. Assurance that in an IP packet, the source address, destination address, and packet payload cannot be maliciously or accidentally modified in transit without detection by the receiver.

Replay protection. A replay sequence number is used to avoid replay attacks. Furthermore a replay sequence number window is defined and only packets whose sequence numbers were within such a window are accepted.

Confidentiality. Assurance of data privacy. This ensures that only the intended receiver is able to decrypt the received data.

The IPsec uses AH and ESP to provide various combination of security services. The AH protocol provides authentication for connectionless integrity, data origin authentication and (optionally) replay protection. Instead, ESP ensures confidentiality, data origin authentication and data integrity, anti-replay service.

Since CL-IPsec has been tailored towards the AH protocol, alongside security services provided by standard AH, CL-IPsec ensures partial integrity (only the UDP-Lite sensitive part) and data origin authentication.

IV. CL-IPSEC IMPLEMENTATION

CL-IPsec is applied to an outbound packet only if there is an active security association (SA) between end-systems exchanging the packet. Then, the first step is to query the security policy database (SPD) to find the applied policy on the outgoing packet. If the packet must be processed (CL-IPsec applied) then either a SA exists, and so the SA is retrieved from the security association database (SAD), or the SA does not exist, and thus a new SA must be created. If the SA is retrieved the system gets the mode to be applied. CL-IPsec is tailored only for transport mode and the AH protocol.

Figure 2 shows how AH is applied to an outbound packet in CL-IPsec.

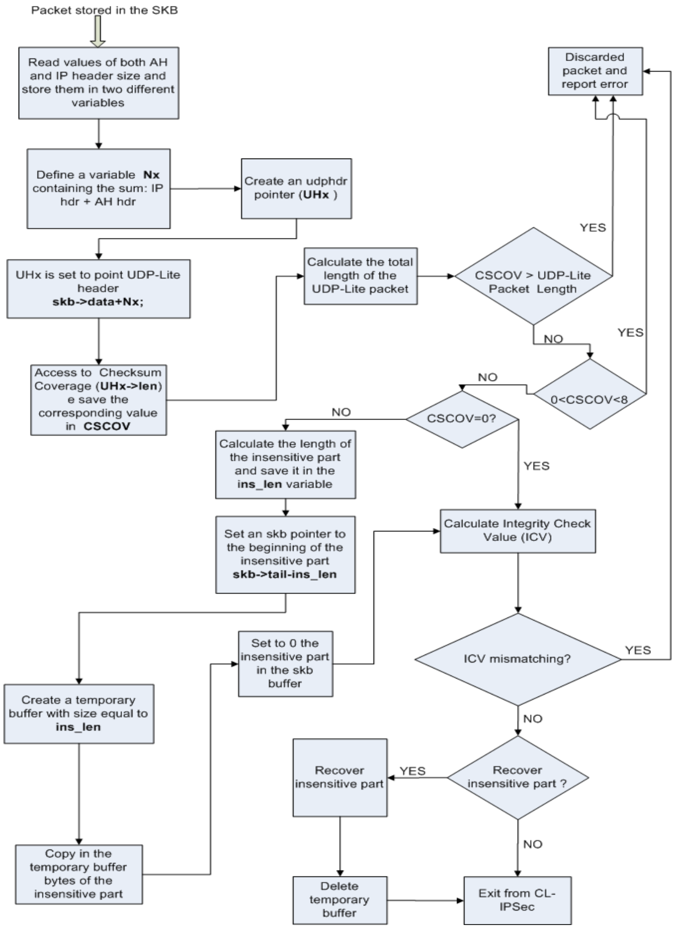
Packet stored in the SKB
Read values of both AH and IP header size and store them in two different variables
Discarded packet and report error
Define a variable Nx containing the sum: IP hdr + AH hdr
Create an udphdr pointer (UHx)
YES
UHx is set to point UDP-Lite header skb->data+Nx;
Calculate the total length of the UDP-Lite packet
CSCOV > UDP-Lite Packet Length
NO
YES
Access to Checksum Coverage (UHx->len) e save the corresponding value in CSCOV
NO
0<CSCOV<8
NO
CSCOV=0?
Calculate the length of the insensitive part and save it in the ins_len variable
YES
Set an skb pointer to the beginning of the insensitive part skb->tail-ins_len
Calculate Integrity Check Value (ICV)
Set to 0 the insensitive part in the skb buffer
ICV mismatching?
YES
Create a temporary buffer with size equal to ins_len
NO
Recover insensitive part
YES
Recover insensitive part ?
NO
Copy in the temporary buffer bytes of the insensitive part
Delete temporary buffer
Exit from CL-IPSec

Figure 2: CL-IPsec module

A. CL-IPsec outgoing traffic management

CL-IPsec module is run only for an UDP-Lite packet; otherwise normal IPsec actions are performed. The general procedures are as follows:

1. Insert of the AH header in the processed IP packet.
2. Generation of the sequence number, which is set to

0 when an SA is initialized. At each AH processing, this sequence number is incremented and copied into the corresponding field of the AH header.

3. In the case of UDP-Lite packets, the CL-IPsec module is run.
4. An algorithm defined by the SA is used to generate the ICV. If required, the authentication data field is padded.
5. Fragmentation. The IP fragmentation can be applied to the packet only after the AH processing.

B. *CL-IPsec incoming traffic management*

Before processing an incoming IP packet, the packet is reassembled. If the “protocol” field of IP header specifies AH and packet matches an SPD entry, the packet is processed by the CL-IPsec. Then, the IP destination address and the security parameter index (SPI) are used to query the SAD to retrieve the SA.

The general procedures are as follows:

1. Sequence number validation. If the retrieved SA specifies anti-replay protection, the sequence number is checked. If the sequence number was already encountered the packet is discarded, else it is accepted.
2. Store ICV contained in the received packet.
3. In the case of UDP-Lite packets, the CL-IPsec module is run; otherwise the ICV is directly computed through the algorithm specified in the SA.
4. If the received ICV matches the computed ICV, then the packet is accepted.

V. PERFORMANCE EVALUATION

A. *Test Bed description*

CL-IPsec envisages modifications in the *ah4* module of the Linux kernel and is implemented in the kernel release 2.6.20.1. In order to both validate the implementation and to evaluate benefits coming from the proposed CL architecture a satellite emulation platform has been set up. Specifically, 3 PCs have been interconnected, as shown in the Figure 3, with the following configuration:

- *ST1* and *ST2* represent the end-systems of a satellite link: e.g. satellite terminal and a satellite gateway.
- *SAT* represents a transparent GEO satellite and introduces physical delays in both the communication directions and bandwidth constraints.

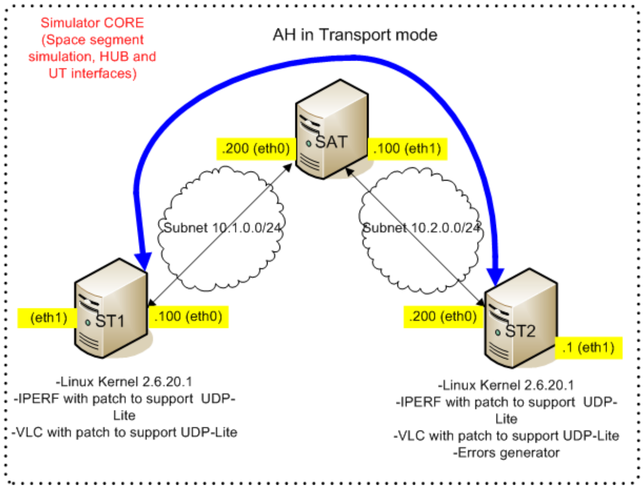

Figure 3: Test bed description

Either a CL-IPsec or an IPsec SA can be established between ST1 and ST2: the security protocol is AH and SPD/SAD databases are manually configured (no rekeying, infinite SA lifetime). UDP-Lite is run as transport protocol (a standard feature since 2.6.11), while both IPERF [20] and VLC [21], patched to run over UDP-Lite, are used as applications.

To emulate an error-tolerant link layer, possible bit errors are generated at the network layer (just before entering CL-IPsec/IPsec routines) of the receiving application end-systems (usually ST2) by using a random error generation with a uniform distribution.

B. *Iperf transfers with UDP-Lite*

Iperf tests consist in the transfer of dummy packets from an iperf server (ST1) to an iperf client (ST2) over UDP-Lite.

Test parameters are summarized in Table 1.

Duration	Packets Length	Bandwith	Checksum Coverage
180 sec	1460 byte	1.05 Mb/s	8

Table 1: Transmission parameters

Both IPsec and CL-IPsec SA were alternatively configured for a set of BER values ranging from 10^{-6} to 10^{-2}. In case of CL-IPsec, IPERF application explicitly set checksum coverage field through the option “-u” followed the desired sensitive part size.

The packet loss rate (PLR) of the two techniques was compared also mathematically under uniformly distributed errors using the aforementioned parameters. Details on the used mathematical model are provided in [22]. The results are shown in Table 2, the subscript “num” are results using mathematical model and “sim” are results obtained from the testbed.

BER	CL-IPsec$_{num}$	CL-IPsec$_{sim}$	IPsec$_{num}$	IPsec$_{sim}$
1.00E-02	98.47%	98.79%	100.00%	100%
1.00E-03	34.05%	35.50%	100.00%	100%
1.00E-04	4.07%	4.30%	70.17%	70.87%
1.00E-05	0.42%	0.36%	11.39%	11.20%
1.00E-06	0.04%	0.02%	1.20%	1.17%

Table 2: Mathematical Analysis of CL-IPsec and IPsec

It can be inferred from Table 2 that the CL-IPsec outperforms IPsec, since packets with corrupt insensitive parts are not dropped. This property of CL-IPsec is also evident from Figure 4. The metric for comparison is packet loss rate perceived versus the BER.

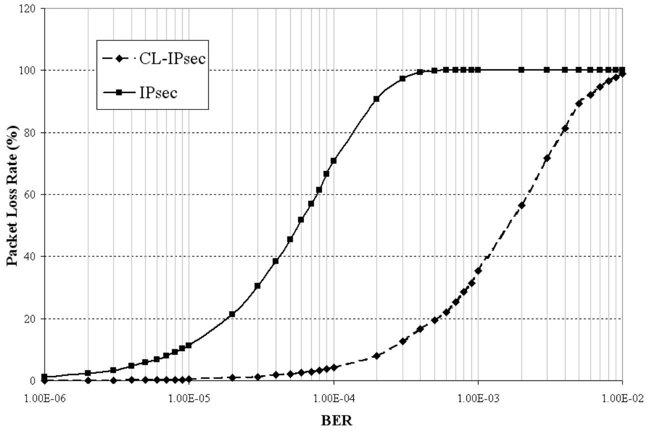

Figure 4: PLR with UDP-Lite/CL-IPSec and UDP-Lite/IPSec

The improvement in packet loss rate offered by CL-IPsec consequently results in an enhancement in performance for error-tolerant applications, while continuing to provide security services.

In Figure 5 the packet loss rate is provided as a function of the Checksum Coverage value ranging from 8 to 45 bytes. Two BER values were considered: 10-^{4} and 10^{-5}.

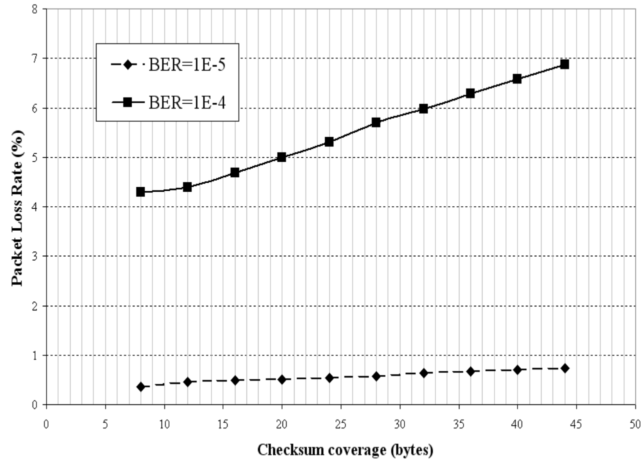

Figure 5: PLR with CL-IPSec for different values of checksum coverage

C. *Video streaming*

Tests were conducted to evaluate the ability of MPEG-2 [13] and MPEG-4 [14] video codec over either UDP-Lite/CL-IPsec or UDP-Lite/IPsec protocol stack to support high quality video streaming even though bit errors affect received packets. Video streaming was between two VLC applications (a sender and a receiver) running over ST1 and ST2 respectively. The UDP-Lite checksum coverage value is defined in the VLC network configuration file. To change it, configuration file must be modified and VLC must be recompiled. In all the considered cases, checksum coverage involves both application and transport header. Test parameters are summarized in Table 3. Specifically, the video streaming duration is 2 minutes and video format is 720x576. BER varies from 10^{-6} to 10^{-3} and the transmission bit rate is set to 1.02 Mbit/s. Both UDP-Lite/CL-IPsec and UDP-Lite/IPsec protocol stacks were alternatively configured.

Streaming Duration	BER	Protocol stack	Codec	Bit rate
120 sec	[10^{-6}-10^{-3}]	UDP-Lite/CL-IPSec	MPEG-2	1.02 Mbit/s
			MPEG-4	
		UDP-Lite/IPSec	MPEG-2	
			MPEG-4	

Table 3: Video streaming: test parameters

Results are shown in the Figure 6 and display the number of dropped frames by application against the BER. Dropped frames are less than 100 overall the BER. However, MPEG-4 outperforms MPEG-2 by reducing the dropped frames by about 20% when using UDP-Lite/IPsec. Finally, CL-IPsec allows a further performance improvement, thanks to an increased number of bytes passed to the application codecs. As a consequence also the performance gap between MPEG-2 and MPEG-4 is reduced.

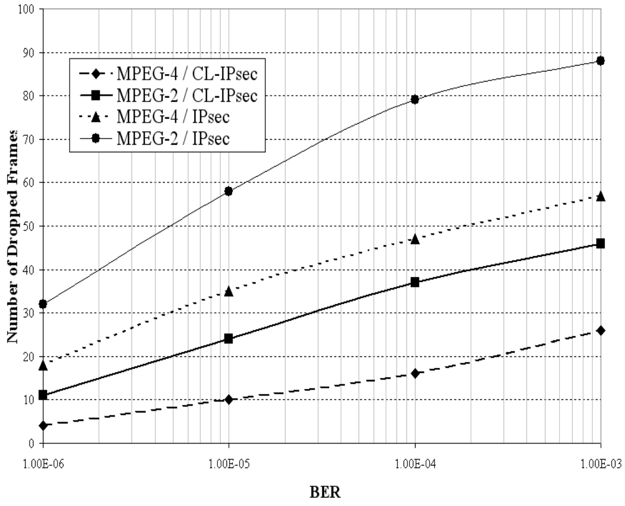

Figure 6: Video streaming with CL-IPsec and IPsec

In Figure 7 and Figure 8 are shown two frames relative to the video streaming tests respectively with MPEG-2 and MPEG-4 codec, when BER=10^{-5}. It is evident that UDP-Lite/CL-IPsec allows a better video quality compared with

UDP-Lite/IPsec. Of course, the rationale is the major number packets that UDP-Lite/CL-IPsec forward to the application, even though corrupted in the insensitive part.

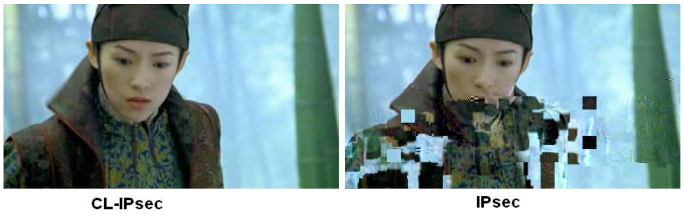

Figure 7: Frames of video streaming test with MPEG-2

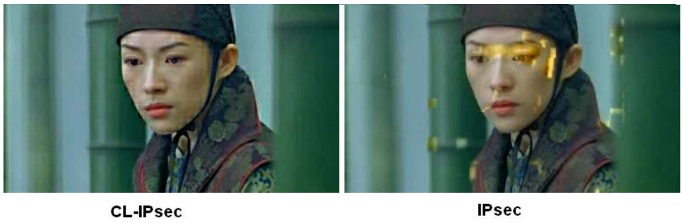

Figure 8: Frames of video streaming test with MPEG-4

These results demonstrate the correct operation of CL-IPsec AH with UDP-Lite and affirm the advantage of using UDP-Lite for error-tolerant multimedia.

VI. Conclusions and Future Work

This paper proposes a Cross-Layer Architecture (CLA) providing network layer security. A cross-layer extension to the IPsec, named CL-IPsec, has been designed, implemented in the Linux Kernel and validated through a satellite emulation platform. Performance results achieved with the proposed CLA are promising and demonstrate that such a technological solution may be very attractive for scenarios affected by bit errors: e.g. UDP-Lite based reliable multicast transport over satellite and video streaming over IP.

Future work will aim to enhance CL-IPsec to support also ESP protocol. In this frame, the main issue is to select an encryption algorithm able to allow that errors in the insensitive part do not affect sensitive part upon decoding. Furthermore, the co-existence of CL-IPsec with security in other layers will be investigated.

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”.

G.Fairhurst also acknowledges support of the project “Satellite Based secure multicast employing hybrid reliability” funded by the Engineering and Physical Sciences Research Council (EPSRC), U.K.

References

[1] W. Stanislaus, G. Fairhurst and J. Radzik, “Cross Layer techniques for flexible transport protocol using UDP-Lite over a satellite network”, 2nd Symposium on Wireless Communication Systems, 2005, pp. 706-710.

[2] G. Giambene (Editor) “Resource Management in Satellite Networks, Optimization and Cross-Layer Design” 2007, Springer, ISBN: 978-0-387-36897-9.

[3] L-A. Larzon, M. Degermark, S. Pink, L-E. Jonsson and G. Fairhurst, “The lightweight user datagram protocol (UDP-Lite)”, IETF RFC 3828, Jul. 2004.

[4] J. Postel, “User datagram protocol”, IETF Tech. Rep. RFC 768, Aug. 1980.

[5] S. Ramachandran, G. Fairhurst, M. Luglio, C. Roseti, S Provenzano, “Network Layer Security: Design for a Cross Layer Architecture”, 2007 International Workshop on Satellite and Space Communication, pp. 271-275, Sep. 2007.

[6] L-A. Larzon, M. Degermark and S. Pink, “UDP lite for real time multimedia applications”, in Proc. IEEE International Conference of Communications (ICC '99), Vancouver, British Columbia, Canada, Jun. 1999.

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

[8] F. Arnal, L. Dairaine, J. Lacan and G. Maral, “Cross-layer reliability management for multicast over satellite”, Computer Networks and ISDN Systems, May 2005, pp. 29-43.

[9] ETSI TS 127 010 v4.2.0, “Digital cellular telecommunications system (Phase 2+) (GSM); Universal Mobile Telecommunications System (UMTS); Terminal Equipment to Mobile Station (TE-MS) multiplexer protocol”, 2002-2003.

[10] K. Seo and S. Kent, “Security architecture for the internet protocol”, IETF RFC 2401, Nov. 1998.

[11] S. Kent, “IP authentication header”, IETF RFC 4302, Dec. 2005.

[12] S. Kent “IP encapsulating security payload”, IETF RFC 4303, Dec. 2005.

[13] L. Chiariglione, “MPEG and multimedia communications”, IEEE Transactions on Circuits Syst. Video Techn., vol.7, pp. 5-18, Feb.1997.

[14] D. Marpe, T. Wiegand, and G.J. Sullivan, “The H.264/MPEG4 Advanced Video Coding Standard and its applications”, IEEE Communication Magazine, Vol. 44,pp. 134-143, Aug. 2006.

[15] M. Handley, et al. “The Reliable Multicast Design Space for Bulk Data Transfer”, RFC 2887, Aug. 2000.

[16] B. Whetten, et al. “Reliable Multicast Transport Building Blocks for One-to-Many Bulk Data Transfer, RFC 3048”,Jan. 2002.

[17] M. Luby, et al., “The use of Forward Error Correction (FEC) in Reliable Multicast”, RFC 3453, Dec. 2002.

[18] T. Paile, M. Luby, R. Lethonen, V. Roca, R. Walsh, “FLUTE – File Delivery over Unidirectional Transport”, RFC 3926, Oct. 2004.

[19] M. Luby, J. Gemmel, L. Vicisano, L. Rizzo, J. Crowcroft, “Asynchronous Layered Coding (ALC) Protocol Instantiation”, RFC 3450, Dec. 2002.

[20] IPERF, URL: http://dast.nlanr.net/Projects/Iperf/

[21] VLC, URL: http://www.videolan.org/vlc/

[22] S. Ramachandran, G. Fairhurst, M. Luglio, C. Roseti, S. Provenzano, “Network Layer Security: Design for a Cross Layer Architecture”, 2007 International Workshop on Satellite and Space Communication, pp. 271-275, Sep. 2007.