Voice over IPsec: Analysis and Solutions

Roberto Barbieri, Danilo Bruschi, Emilia Rosti
Dipartimento di Scienze dell'Informazione
Università degli Studi di Milano
{danilo.bruschi, emilia.rosti}@unimi.it

Abstract

In this paper we present the results of the experimental analysis of the transmission of voice over secure communication links implementing IPsec. Critical parameters characterizing the real-time transmission of voice over an IPsec-ured Internet connection, as well as techniques that could be adopted to overcome some of the limitations of VoIPsec (Voice over IPsec), are presented. Our results show that the effective bandwidth can be reduced up to 50% with respect to VoIP in case of VoIPsec. Furthermore, we show that the cryptographic engine may hurt the perfomance of voice traffic because of the impossibility to schedule the access to it in order to prioritize traffic.

We present an efficient solution for packet header compression, which we call cIPsec, for VoIPsec traffic. Simulation results show that the proposed compression scheme significantly reduces the overhead of packet headers, thus increasing the effective bandwidth used by the transmission. In particular, when cIPsec is adopted, the average packet size is only 2% bigger than in the plain case (VoIP), which makes VoIPsec and VoIP equivalent from the bandwidth usage point of view.

1. Introduction

The Internet community agrees that security is one of the key properties that should characterize any ICT (Information and Communication Technology) system and application, with particular emphasis on those that rely on the Internet for their very nature, e.g., e-commerce. Unfortunateley, security does not come for free and, in general, security and efficiency are conflicting requirements. While for a broad class of Internet applications such a fact has limited effects, there is a class of applications whose functionality may be compromised by security controls, namely real-time applications, i.e., applications that impose time constraints on packet delivery in order to reproduce the original source of information. As an example of such applications videoconferencing, VoIP (Voice over IP), or real-time video can be considered. In all these applications, introducing a layer that guarantees packet content confidentiality, integrity and authentication can slow down packet transmission, which may not be acceptable by the application itself.

In this paper we present the results of the experimental analys of the transmission of voice over secure communication links implementing IPsec. The goal of our investigation is to understand whether the current VoIP applications can simply be replaced by VoIPsec (Voice over IPsec) once IPsec has become widely deployed. We adopted an experimental approach in order to be able to actually quantify the critical parameters for the VoIP application (e.g., delays introduced by routers internal computations, queueing delays experienced by packets when waiting to be routed). Experiments were conducted in order to estimate such parameters when voice is transmitted over an IPsec channel. Based on these observations, we propose a compression scheme for IPsec packet headers.

Various aspects have to be considered in order to address the problem of real-time transmission over secure channels, in particular VoIPsec. The real-time nature of the problem poses some constraints. In the case of voice transmission, the maximum acceptable delay in packet delivery for optimal voice quality is 150ms, which can be extended up to 200ms in case of encrypted communications. Thus, in a standard VoIP application, after the signal has been digitized, there are 150ms to code the signal using some standard scheme [e.g., ITU standards G.729, G.723, etc.], divide it into packets and encapsulate the packets into IP packets, then route the packets on the Internet, and reconstruct the original traffic stream at the destination, where it usually is buffered in

order to smooth the jitter. Because of such a timing constraint, voice packets are small (10-50 bytes long payload) in order to guarantee that all the above mentioned operations can be performed within the given time constraint. When IPsec is considered, the in packet size increases due to the IPsec specific headers (ESP, AH, new IP header for tunnelling) added to each packet. This significantly increases the ratio of header size to payload size, thus reducing the effective bandwidth, i.e., the percentage of bandwidth carrying actual data w.r.t. the total bandwidth used. Furthermore, the time necessary to build such headers and to apply the necessary cryptographic functions to the payload introduces additional delay to packet trasmission.

Our results show that the effective bandwidth can decrease up to 63% in case of VoIPsec w.r.t. VoIP, making VoIPsec inadequate for low bandwidth connections (e.g., via modem). The most interesting result is, however, related to the cryptographic engine, or crypto-engine. It is not surprising that for voice traffic the crypto-engine can be a serious bottleneck. What is not immediately evident is the actual rational of such a behavior. Rather than the expected constraints on the crypto-engine throughput, the critical factor turned out to be the impossibility to control and schedule access to the crypto-engine so as to favor real-time traffic over regular one. This applies regardless of whether the scheduler is implemented as a software module or a hardware component. Therefore, if voice traffic is interleaved with other types of traffic, e.g., ftp or http traffic, during a secure session, it may happen that the latter (usually characterized by big packets) is scheduled in the crypto-engine before voice traffic. In this case voice traffic might be delayed to the point that packets are discarded most of the times.

Finally, we present a new compression scheme that allows to improve the effective bandwidth used by VoIPsec applications. Our compression scheme reduces the size of the internal headers of a voice packet, based on the observation that some of the information they carry does not change. By adopting our compression scheme, VoIPsec packets are only 2% longer than regular VoIP packets, rather than 50% longer plain VoIPsec packets.

This paper is organized as follows. Section 2 presents a quick overview of VoIP. Section 3 describes the testbed used for our experiments. Section 4 presents the experimental results of the analysis of VoIP over IPsec. Section 5 describes the proposed compression scheme and the measured performance when it is adopted. Section 6 concludes the paper and summarizes our findings.

2. Preliminaries

In this section we describe the problems occurring when voice is transmitted on a computer network and how they change when functions that guarantee confidentiality and authentication of the communication are introduced, namely IPsec. The technologies involved are:

- VoIP, the application for digitizing, compressing and converting voice into IP packets, and transmitting them over IP networks;
- IPsec, the module for encryption and authentication of information at network layer;
- QoS protocols, that ensure voice quality when it is transmitted over IP networks.

In the following we briefly recall the basic techniques involved in each technology.

2.1 Voice over IP

VoIP is a technique for transmitting voice data over the Internet [2, 4]. The following steps are performed:

- digitization of the analog signal, usually performed at a frequency of 8 KHz with 8 bit per sample, thus generating 64Kbytes per second;
- packet generation of the digital signal according to the TCP-UDP/IP protocols;
- transmission of the packets on the network;
- packet reception and analog signal reconstruction at the destination.

When sending voice traffic over IP networks, a number of factors contribute to overall voice quality as perceived by an end user. Some of the most important factors are end-to-end delay in the voice carrier path and degraded voice quality. Among the factors that degrade voice quality are packet loss, delay variation, or jitter, voice compression schemes, transducers (microphones and speakers), echo cancellation algorithms, and voice activity detection at voice endpoints. In this paper we focus on end-to-end delay and packet loss.

VoIP is a typical real-time application as the original signal has to be reproduced at the destination as close as possible to the instant when it was generated, therefore the signal delay is a qualifiying parameter for VoIP application. Each of the steps mentioned before introduces some delay in packet transmission. ITU G. 114 recommendation suggests to contain such a delay within a limit of 150 ms for wired communication lines and within 250-300 ms for satellite based communications. Various factors influence signal delay during a VoIP transmission. The time spent by the CODEC, the device that performs the digitization process, may vary between 0.75-30ms, depending on the coding schemes adopted and the quality of the reproduced signal. The queueing delay (i.e., the time spent by a packet in the router buffers waiting for being routed) may add up to 30 ms. A further delay in the range of 40-70ms, called jitter delay, is introduced by buffering arriving packets so that they can be delivered at a uniform rate. Buffering is necessary to eliminate the variation of

the delivery rate caused mostly by queueing time due to network load.

Table 1 reports the number of phone calls (using VoIP) that can be performed with up to date technology given channels with different bandwidth and different payload per packet.

	10		20		40	
B/w	#calls	delay	#calls	delay	#calls	delay
32	0	-	0	-	1	>200
64	0	-	1	100-150	2	150-200
128	1	<100	2	~100	4	150-200
256	2	<100	5	~100	9	~150
512	5	<100	10	<100	18	100-150
1024	11	<100	20	<100	36	100-150
10240	117	<100	214	<100	365	~100

Table 1: Number of telephone calls and average delay in ms as a function of channel bandwidth (B/w, from 32Kbps to 10 Mbps) and payload size (10, 20, and 40 bytes).

Table 2 reports the main characteristic in terms of bit rate (in Kbps), compression delay (in ms) and Mean Opinion Score (MOS) for a set of algorithms that can be adopted by CODECs [2]. The MOS is a parameter used to measure the quality of the signal reproduced by such algorithms and ranges from 1 to 5, 1 being the worst case. As the table shows, the best algorithm is significantly better than the rest, which obtain a MOS in the surrounding of 3.5.

Compression Algorithm	Bit Rate [Kbps]	Delay [ms]	MOS
G.711 PCM	64	0.75	4.1
G.726 ADPCM	32	1	3.85
G.728 LD-CELP	16	≤ 5	3.61
G.729 CS-ACELP	8	10	3.92
G.729a CS-ACELP	8	10	3.7
G.723.1 MP-MLQ	6.3	30	3.9
G.723.1 ACELP	5.3	30	3.65

Table 2: CODEC algorithms and their characteristic bit rate in Kbps, compression delay in ms, and MOS.

With VoIP, voice is sent to the destination using an RTP (Real-time Transport Protocol) that operates above UDP [4]. A TCP connection between the two peers is used to set up and tear down calls, to negotiate capabilities and to set up the RTP channel, which is then used to transmit data. To guarantee the security of the communication, both the TCP and the RTP channels have to be encrypted and authenticated. A reasonable choice to implement such functionalities is IPsec.

2.2 IPsec

IPsec [6] provides security services for IP traffic by allowing a host to set up a secure IP channel with any peer it wishes to connect to. The host can choose different services depending on the level of security required. The services provided by IPsec are based on two protocols: an authentication protocol (AH) and a combined encryption and authentication protocol (ESP). The first protocol provides services such as connectionless integrity and sender authentication, while the second protocol is in charge of guaranteeing confidentiality among other services.

In order to execute any of these algorithms both the peers have to have previously agreed on pairs of secret keys. Such an agreement is performed by the IPsec key management module implementing the ISAKMP/Oakley protocol. Such a module mutually athenticates the peers, then it negotiates the symmetric keys they need to exhange messages and the cryptographic algorithms they willl use. IPsec encodes the information needed to perform AH and ESP services in two additional packet headers called AH and ESP headers, respectively.

IPsec may operate in two different ways, depending upon whether the secure communication is between two endpoints directly connected (in which case it operates in transport mode) or between two intermediate gateways to which the two endpoints are connected via a clear, i.e., unencrypted, channel (in which case IPsec operates in tunnel mode). For voice applications the following considerations apply: confidentiality is essential, authentication "in band" is expensive, session endpoints usually are not the cryptographic endpoints. Therefore, the best choice to secure voice traffic is to use the ESP header in tunnel mode.

2.3 Quality of Service

When executing VoIP applications, QoS protocols must be adopted in order to be able to meet the requirements on transmission parameters such as transmission delay, jitter and buffering delay [3,9]. QoS protocols try to meet the imposed requirements using different features such as packet classification, queueing mechanisms, traffic shaping, header compression, congestion avoidance

strategies and Resource Reservation protocols. Unfortunately, such features cannot be taken advantage in combination with IPsec, as they use fields in the IP header that IPsec encrypts. Thus, when IPsec is used, the possible choices of QoS protocols are limited.

In our experiments we consider the following QoS protocols: Diff-serv environment with TOS/DSCP marking (Type Of Service/Diff-Serv Code Point) to manage congestion and packet discard, the LLQ (Low Latency Queueing) queue management protocol to handle RTP packets properly, and the LFI (Link Fragmentation and Interleaving) protocol for packet fragmentation, which interleaves fragments with voice packets [10]. Note that in a diff-serv environment it is necessary to copy the DSCP field in the external IP header, so that the QoS mechanism can recognise the packet priorirty and treat it accordingly.

3. Experimental environment

In this section we describe the environment where most of the experiments described in Section 4 and 5 were performed. As Figure 1 shows, three routers (R1, R2, and R3) with two independent IPsec tunnels (IPSEC 1 and IPSEC 2 in the figure) are between the two telephones. A fourth routers (TGN+PKTS) operates as traffic generator up to a rate that may lead the links and the crypto-engines to saturation. The various subnetworks parameters and types of routers are reported in the figure.

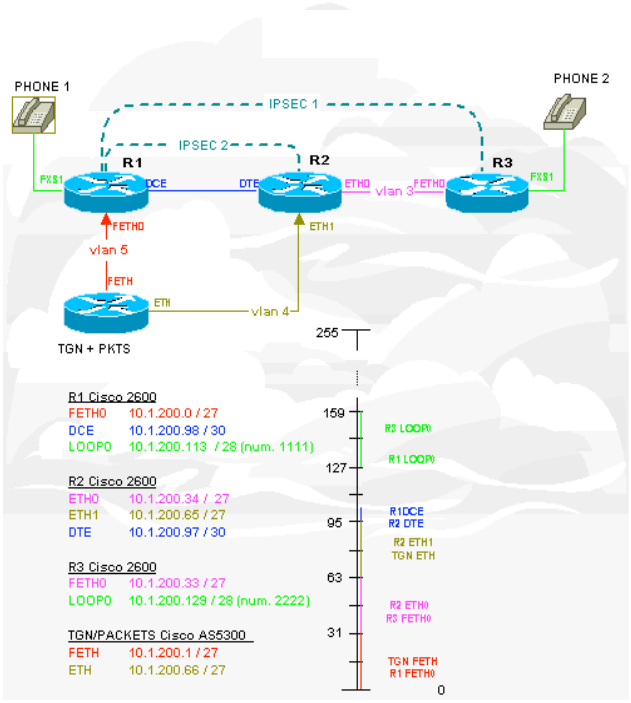

Figure 1: Experimental environment.

We configured the following QoS features:

- the dial peers set the TOS bit for the signaling and the media flows (IOS ver. 12.2);
- the LLQ protocol is set with a reserved bandwidth of 64Kbps on the serial link and on the Ethernet link;
- the serial link is a PPP multilink with LFI enabled and with maximum latency set to 10ms;
- RTP addresses are forced to match the access lists.

Different tests have been performed in order to estimate various parameters such as effective bandwidth usage, impact of various QoS strategies on traffic delay, crypto-engine throughput, impact of various encryption algorithms on packet delay. The complete set of results is reported in [1]. For the sake of brevity, we only report here the most significant results.

4. IPsec and voice transmission

Two main factors affect voice traffic when IPsec is used. The first one is the increased packet size because of the headers added to the original IP packet, namely the ESP header for confidentiality and the new IP header for the tunnel. The second one is the time required to encrypt payload and headers and the construction of the new ones.

In this section we report the results obtained in measuring the influence of such factors on voice traffic. Realistic estimates of such factors can be determined only through a careful experimental analysis, as most of the parameters involved such as traffic shape, buffering delay and queueying dealy depend on real traffic condition.

4.1 Packet size

In Figure 2 the format of voice packets with various protocols is illustrated for a 40 bytes payload, a typical packet length for voice traffic. The picture shows how packet format and size change with and without IPsec and for various combinations of cryptographic algorithms. The overall minimum size is obtained when compressed RTP (cRTP) is adopted (second bar from the bottom) in which case the header size is only 20% of the payload size, yielding a 45 bytes long packet, while a regular IP packet is 80 bytes long. As it can be seen, the use of IPsec dramatically increases the size of the packets, which reaches 120 and 130 bytes depending on the cryptographic services requested. As a consequence, the ratio between the actual payload and the total packet length decreases, indicating an increase in “wasted” bandwidth, i.e., bandwidth that does not carry actual data.

Table 3 reports similar information in details, with the ratio of the payloadt to total packet length (fourth column from the right), the relative size increase w.r.t. cRTP (third column from the right), and the performance reduction in terms of bandwidth usage w.r.t. cRTP (second righmost

colum). In order to give an immediate perception of the effective bandwidth reduction, in the righmost column we also report the number of phone calls possible for each protocol type for a 128Kbps line with 50 packets per second (pps). As the figures show, if IPsec is used, the number of phone calls is half that with IP, or about one third of the number of calls possible when cRTP is used [5,8].

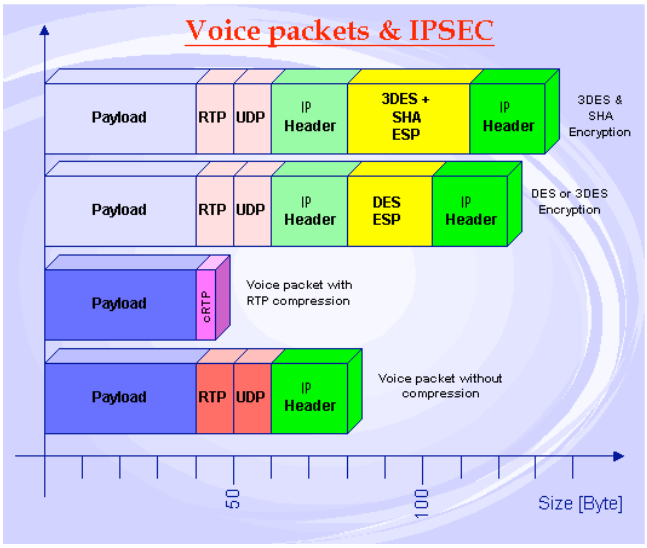

Figure 2: The format of voice packets with IPsec (top two bars) and with IP (bottom two bars) for a 40 bytes long payload.

Packet Type	Hdr [Byte]	Pkt len. [Byte]	Ratio	Size incr.	Perf. Reduc.	#call
cRTP	5	45	89%	0%	0%	7
IP	40	80	50%	78%	44%	4
IPsec DES	82	122	33%	171%	63%	2
IPsec 3DES+SHA	94	134	30%	198%	66%	2

Table 3: Header and total packet length, payload to packet lenght ratio, relative size increase, performance reduction and number of phone calls on a 128Kbps and 50 pps link for 40 bytes payload packets and various protocols.

We now investigate the impact of different encryption algorithms to encrypt the payload on the packet size. We consider the combinations of DES, 3DES, NULL and SHA [12] for authentication and 3DES and SHA only. Figure 3 reports the percentage increase in packet size as a function of the original packet size for DES, 3DES, NULL+SHA (bottom line) and for 3DES+SHA (top line). As the figure shows, the impact of different encryption algorithms is negligible, especially as the packet size increases.

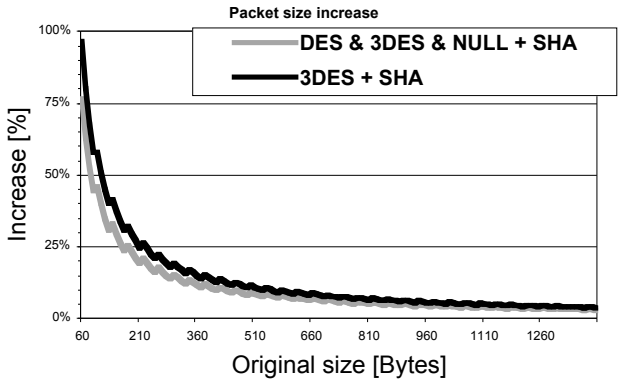

Figure 3: Packet size increase for two sets of cryptographic functions, as a function of packet size in bytes.

The packet size increase has negative effects not only on bandwidth usage but it also impacts on the transmission delay, router internal delays, queueing delay, thus affecting jitter and overall packet delay. The transmission delay increases proportionally with the packet size and is constant for every router (whether peers or not). Internal router delays (e.g., due to checksums calculation) are considered in the generic IPsec delay. Queueing delay is sensitive to packet size as well and this is evident with low bandwidth links.

In order to evaluate such parameters we injected multiple traffic streams in our test network, which start at random times in order to create a realistic scenario. Individual flows may be distinguished based on the IP source address, which allows to study links with critical traffic (close to the congestion rate) and also to evaluate the impact on the system determined by the queueing delay. The results of these tests are summarized in Figure 4, where the measured traffic delay is reported as a function of the traffic intensity in pps on a 128 Kbps link with 90 bytes long packets. Two sets of curves are graphed, the leftmost ones referring to DES encrypted packets and the rightmost ones relative to plain transmission. For each of the two cases, three curves are reported. The dashed ones were obtained by solving a queueing network model, the ones with a sharp turn at saturation are relative to the ideal case with no queueing, and the intermediate ones are the actual measured delays.

As Figure 4 shows, in case of encrypted traffic (leftmost set of curves), the traffic delay grows much earlier (i.e., for smaller traffic rates) than in case of clear traffic. Such a result is not surprising since the time requested by encryption cannot be neglected. It is

interesting to note that the theoretical results capture the measured data fairly well in case of clear traffic but they are not as accurate in case of encrypted traffic. The reason is that it is not possible to estimate a priori the time spent by a packet before it accesses to the crypto-engine. In real life scenarios, such a phenomenon is more evident since packets often traverse a number of routers greater than three, as we had in our experiments.

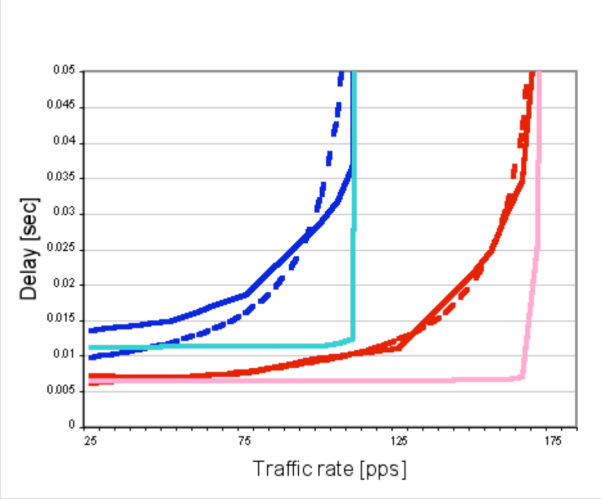

Figure 4: Packet delay in s for clear (rightmost curves) and DES-encrypted (leftmost curves) traffic as a function of traffic rate in pps on a 128Kbps serial link with 90 bytes long packets. Dashed lines are the queueing theory predicted value, smooth slod lines are the measured values, and sharp turning lines are the ideal result in case of no queues.

4.2 Crypto-engine

During the encryption process, a router performs some operations, namely packet encryption and new headers construction (ESP + IP tunnel), that influence the CPU utilization and introduce a further delay. We performed a series of experiments in order to evalute the impact of such parameters on the transmission of voice traffic. Our experiments show that the crypto-engine is a serious bottleneck in the transmission of real-time traffic in IPsec. The main reason, however, is not the low efficiency of the encryption process but the impossibility to control packet access to the crypto-engine. While we can use QoS protocols to speed up the routing phase, there is no way to indicate a priority in the packets in order to modify scheduling choices of the crypto-engine. In what follows we report the results of the experiments performed in order to evaluate the perfomance of the crypto-engine. The encryption delay and throughput were measured.

In order to measure the maximum encoding rate, when different algorithms are used, we performed the following experiments. We considered the cryptographic algorithms DES, 3DES, NULL & SHA, 3DES & SHA (all implemented in software) and for each case we generated 4 packet flows with packets of size 60, 100, 250, 1000 Bytes, respectively. Each flow starts from 0 pps and increases its rate of 25 pps every 30 s in order to saturate the crypto-engine. In order to avoid the effect of possible link saturation, we performed all these experiments with a 100Mbps link. Results are the reported in Figure 5 and 6.

Figure 5 graphs the measured throughput as a function of the global traffic flow. The straight line is throughput for transmission of packets in the clear, therefore it increases linearly with traffic. The figure shows that when encryption is performed, throughput levels off or decreases after reaching a maximum value, which depends on the algorithm. The best performance is achieved by DES, then 3DES, NULL+SHA, and last is the 3DES+SHA case. The figure shows that, as expected, the lighter the computation, the higher the throughput value achieved. It is interesting to note that even in the best case, i.e., for DES, such a maximum value is only slightly above 1000 pps. The negative slope throughput exhibits after reaching the maximum is due to packets discarded by the engine because it is saturated. The graphs in Figure 5 are particularly important for our VoIP and VoIPsec comparison, since discarded packets contribute to lower the quality of the signal during the reconstruction phase.

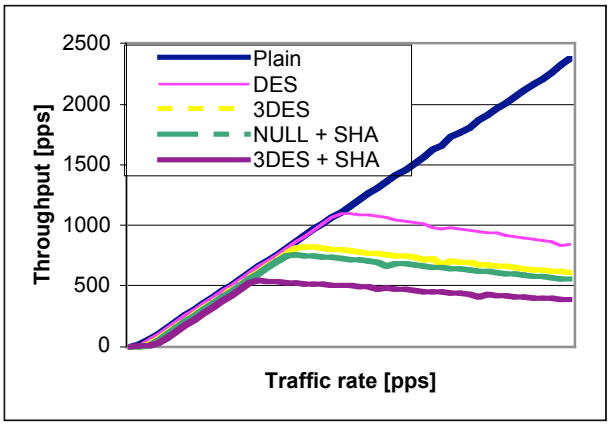

Figure 5: Throughput of the crypto-engine in pps as a function of linearly increasing traffic in pps for plain and encrypted traffic.

Figure 6 illustrates the crypto-engine throughput in Mbps for each cryptographic function set for the various packet sizes, namely 60, 100, 250 and 1000 bytes (left to right bars in each group). As the figure shows, in case of DES and NULL+SHA longer packets significantly improve the crypto-engine performance. A similar trend, but with a much less dramatic effect, is observed with

3DES+SHA. Slightly anomalous is the behavior with 3DES only, although the absolute throughput values are in the same range as those measured for 3DES+SHA. The longer and more elaborate computation of 3DES is responsible for the lower throughput. Therefore, from a qualitative point of view, the crypto-engine behaves as a link with bandwidth equal to its maximum throughput.

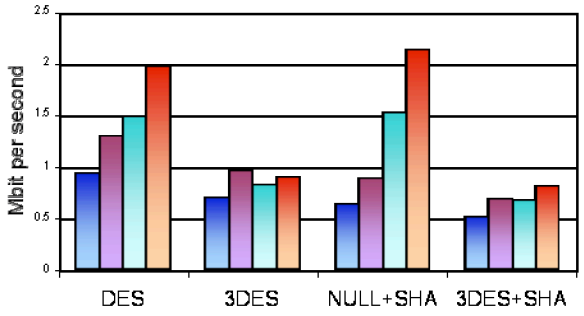

Figure 6: Crypto-engine throughput in Mbps for 60, 100, 250, and 1000 bytes long packets (bars from left to right in each group) for various sets of cryptographic functions.

We then performed experiments with real voice traffic, instead of a syntethic flow of packets. We considered a 3DES encrypted phone call without any other traffic on the link, whose results are reported in Figure 7, and a 3DES encrypted call on a link carrying 1200 byte packets extra traffic, whose results are reported in Figure 8. In both cases, the interarrival time between consecutive packets is plotted. The spike in Figure 7 is due to a single late packet, which was immediately followed by the next one.

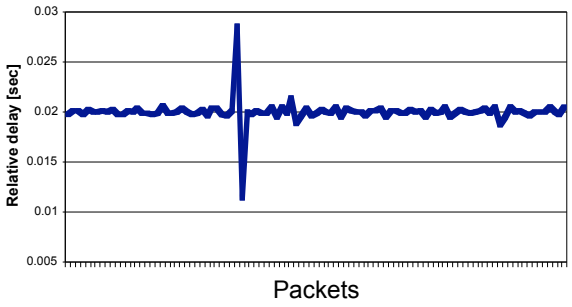

Figure 7: Individual packet interarrival time for voice traffic on an empty link.

As the figures show, the two graphs differ significantly, thus showing how much voice traffic suffers from heterogeneous traffic on the link.

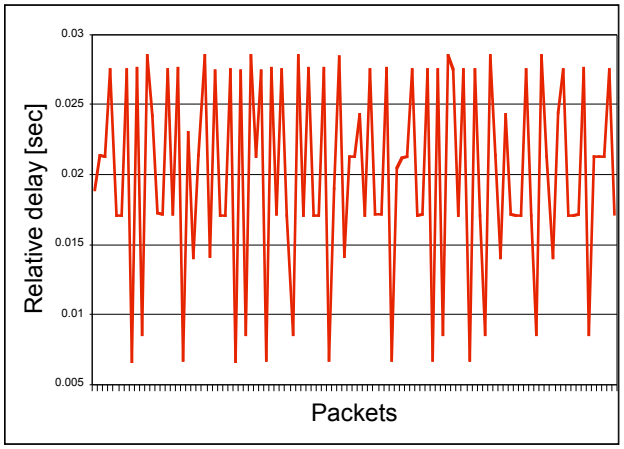

Figure 8: Individual packet interarrival time for voice traffic on a link with 1200 bytes long packets extra traffic.

4.3 QoS and VoIPsec

We now describe a test aimed at verifying the importance of QoS protocols during the transmisison of encrypted voice traffic. For this experiment we generated three traffic streams: a phone call (with the TOS bit set) with 70 bytes long packets at a rate of 50 pps (T.Stream 1), a phone call without priority and equal parameters (T. Stream 2), and an extra stream of jumbo datagrams with 1500 bytes long packets at a rate of 1pps (T. Stream 3), which simulates ordinary traffic. All traffic streams are 3DES encrypted. Traffic characteristics are summarized in Table 4.

T.Stream	Length	Rate	TOS
1	70	50	Yes
2	70	50	No
3	1500	1	No

Table 4: Traffic stream characteristics for the QoS experiments. All streams are encrypted with 3DES.

For all types of traffic we measured the traffic delay as the difference between arrival time of consecutive packets. As it can be seen from the data reported in Figure 9, calls performed without QoS control (T. Stream 2, central bar in each set) suffer a great variability in the measured interarrival times, which accounts for the largest standard deviation. The more stable stream is the one comprised of jumbodatagrams (T. Stream 3, rightmost bar in each set), which is less sensitive to variations. The phone call without priority (T. Stream 1, leftmost bar in each set) experiences some variability, although not as significant as the phone call with no priority.

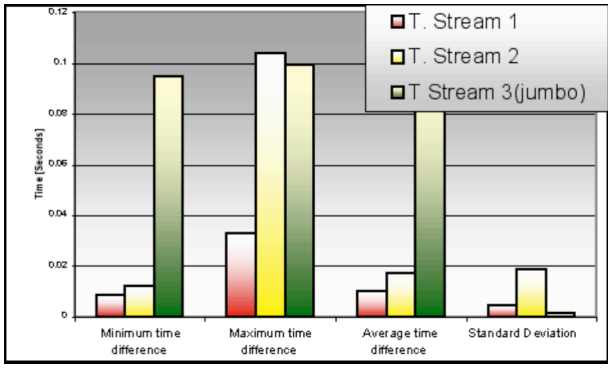

Figure 9: Minimum (leftmost), maximum (second leftmost), and average (third leftmost) packets interarrival time with std. deviation (rightmost) for three traffic streams (T. Stream 1: leftmost bar in each group, T. Stream 2: central bar in each group, T. Stream 3: rightmost bar in each group) with (T. Stream 1) and without QoS (T. Stream 2 and 3).

5. Header compression

As our experiments show, one of the most critical aspects in transmitting voice traffic over networks that implement IPsec is the increase in packet size due to IPsec itself. In this section we describe a header compression scheme that could be adopted to solve such a problem and present measurements of the proposed scheme.

5.1 cIPsec

The idea is based on the observation that most of the fields belonging to the internal headers of a packet, i.e., IP original header, UDP and RTP headers carry values that either remain constant over the entire life of the connection, or change but the second order difference, i.e., the difference of consecutive differences computed on packet parameters, is zero. Furthermore, some other fields contain information already present in the external headers. Such an approach is adopted also in low speed serial link compression (cSLIP), although it applies to TCP/IP flow rather than UDP [13].

Starting from such observations, we devised a compression scheme that allows to reduce the internal IP/UDP/RTP headers down to four bytes for most packets. A routine was added to the IPsec module at the operating system level. Such a routine is composed of two modules, a compressor module that performs packet compression on outgoing packets and a decompressor module that restores the headers original content of incoming traffic. Since the idea is based on the cRTP [5, 8] protocol, we call our compression scheme for IPsec cIPSec.

cIPsec needs to maintain a collection of shared information and a session context in a consistent state between the two endpoints. Such information is mainly used at the receiving endpoint to reconstruct the original headers. A session context is defined by the combination of the IP source and destination addresses, the UDP source and destination ports, and the RTP SSRC field. The information shared in each context, which is stored in each crypto-end-point, consists of the following items:

- full IP, UDP and RTP headers of the last packet sent by the compressor or reconstructed by the decompressor;
- last value of the 4-bit sequence number, which is used to detect packet loss between the compressor and decompressor.

If the information above described is maintained at the two end-points then during the transmission the IP-UDP-RTP internal headers can be reduced to the header depicted in Figure 10. The meanings of the various fields in Figure 10 are:

- Session Context ID (CID): it identifies the peer to peer communication. These 16 bits allow to maintain up to 65536 calls between the two endpoints. In case of multiple calls, the CID bits are used to index the context information.
- Link sequence: it contains the least significant 8 bits of the RTP header sequence number. It can keep track of up to 256 consecutively lost packets.
- Sequence bit (S): it notifies the presence of the RTP Sequence bits.
- Checksum bit (C): it notifies the presence of the UDP checksum bits.
- UDP Checksum (opt.): it contains the checksum value, sometimes needed to guarantee end-to-end data integrity. This field is optional and can be disabled if the ESP protocol uses an authentication algorithm.
- RTP Sequence (opt.): it is the complete RTP bit sequence used for context synchronization.
- Retransmission bit (R): it indicates the necessity to retransmit and uncompress a packet.

SESSION CONTEXT ID	LINK SEQ.	S	C	R	NOT USED
UDP CHECKSUM (if C =1)					
RTP SEQUENCE (if S or S' =1)					

Figure 10: Compressed IP/UDP/RTP headers.

The main problem with the cIPsec compression scheme is related to the occurrence of transmission errors as they can be detected by the CRC or by the checksum value in

the IP/UDP headers, or because of packet loss. Because of the real-time nature of voice transmission, no error correction is possible and packets are lost. In this case a resynchronization process is required.

5.2 cIPSec performance

In this section we report a preliminary evaluation of the performance of the cIPSec compression scheme. Figure 11 reports the bandwidth used in case of (from left to right in the figure) plain VoIP packets, VoIPsec packets and cIPsec packets for various error rates. As the figure shows, the difference of bandwidth usage in case of VoIP packets and cIPsec packets are comparable, even with error rates as high as 10% (rightmost bar in Figure 10), which force retransmission.

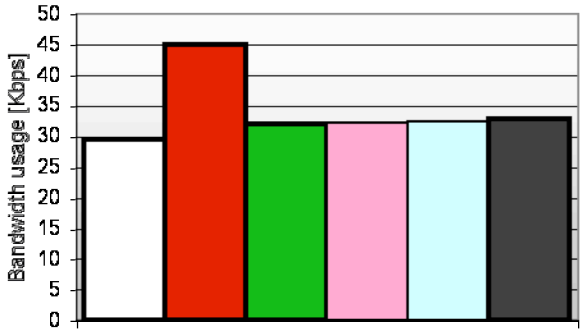

Figure 10: Bandwidth usage for plain IP, plain IPsec and cIPsec: from left to right, plain IP, plain IPsec, cIPsec. The three rightmost bars compare the performance of cIPsec with 2%, 5% and 10% traffic error rate.

The advantages of the adoption of cIPsec when dealing with voice traffic are a better end-to-end bandwidth utilization, reduction of transmission delay, and better usage of the CPU and of the crypto-engine. The expected reduction on the transmission delay is similar to the bandwidth usage optimization and, for two 64Kbps access links, it can be improved up to 10ms. The reduction of the delay due to encryption depends on the algorithm chosen. Relative gains are reported in Table 5.

On the other side, its implementation will negatively impact on the CPU performance and on the memory use because of the additional computational burden put on the CPU by cIPsec. This factor can be estimated by considering the time required for computing a cRTP header, which is a more complex operation than that required by cIPSec and thus the cRTP value can be used as an upper bound for cIPSec. In case of cRTP, our results show an increase in the CPU usage by 4%. As the improvements on the encryption phase due to the use of cIPsec are between 4% to 6,5% it turns out that overall cIPSec does not influence the computation time.

Encryption algorithm	Relative Gain
DES	4%
3DES	6.5%
NULL+SHA	4.4%
3DES+SHA	1.8%
AES128	5.2%
SEAL	4%

Table 5: Relative gains for various cryptographic algorithms when cIPsec is used w.r.t. standard IPsec.

6. Conclusions

In this paper we have presented an experimental study of VoIP when an IPsec network is used. The impact on performance, with particular attention to bandwidth usage and transmission delay has been evaluated through a series of experiments on a real testbed. The critical role played by the scheduling algorithm adopted by the IPsec crypto-engine in affecting real-time traffic delay has been identified. A new compression scheme based on cRTP to improve the effective bandwidth used by secure traffic has been proposed and preliminary performance results presented. Experiments with other types of real-time traffic to see whether the results presented in this paper can be generalized to all real-time traffic are part of future work.

7. Bibliography

[1] R. Barbieri, Studio e analisi delle problematiche di trasmissione di voce cifrata in ambienti con QoS, Ms.Thesis, Univeristà degli Studi di Milano, 2002.

[2] U. Black, **Voice over IP**. Pretice Hall, 1999.

[3] C.-N. Chuah, Providing End-to-End QoS for IP-based Latency-sensitive Applications. Technical Report, Dept. of Electrical Engineering and Computer Science, University of California at Berkeley, 2000.

[4] M. Goncalves, **Voice over IP Networks**. McGraw-Hill, 2000.

[5] M. Leelanivas, RTP Header Compression. Cisco Systems, 1997.

[6] P. Loshin, **Big Book of IPsec RFCs: Internet Security Architecture**. November 1999.

[7] M. Marjalaakso, Security Requirements and Constraints of VoIP. Technical Report, Dept. of Electrical

Engineering and Telecommunications, Helsinki University of Technology, 2001.

[8] D. Nguyen, J. Lequang, cRTP Performance Enhancement. CISCO SYSTEM [ENG 102721].

[9] A. Sayeed, RSVP & its use for Voip. Cisco Systems.

[10] B. Thompson, Voice over IP Quality of Service Architecture and Performance Requirements. Cisco Systems, ENG- 53391, May 2000.

[11] Z. Wang, Internet QoS: Architectures and Mechanisms for Quality of Service.

[12] B. Schneier. **Applied cryptography: Protocols, Algorithms, and Source Code in C**. John Wiley & Sons, Inc., 2nd Edition, 1996.

[13] V. Jacobson, Compressing TCP/IP Headers for Low-Speed Serial Links, RFC 1144, February 1990.