Energy-Efficient Security for Voice over IP

Hoseb M. Dermanilian, Farah Saab, Imad H. Elhajj, Ayman Kayssi, and Ali Chehab
(Corresponding author: Imad H. Elhajj)

Department of Electrical and Computer Engineering, American University of Beirut
Riad El-Solh, Beirut 1107 2020, Lebanon
(Email: ie05@aub.edu.lb)

(Received Feb. 24, 2013; revised and accepted Aug. 14, 2013)

Abstract

The fast spread of handheld smart devices contributed to the development of VoIP softphones running over such devices. Most security mechanisms were mainly designed for desktop PCs and hence did not take into consideration the power constraints of handheld devices. This fact highly motivated the development of new security mechanisms that try to minimize the energy consumption without compromising the security of the exchanged data. In this paper, we propose an energy-efficient security algorithm for VoIP applications running on mobile devices (SecVoIP). The algorithm resolves several weaknesses available in current algorithms while maintaining an appropriate security level. Several experiments were conducted and the results showed significant improvement in processing time, CPU cycles, and consumed energy as compared to SRTP, one of the most widely used security protocols for VoIP. Moreover, we present the results of extensive experimental work that demonstrates that known plaintext attacks against audio streams are not feasible.

Keywords: Energy efficiency, handheld devices, selective encryption, SRTP, VoIP security

1 Introduction

Voice over Internet Protocol (VoIP) has become widely used worldwide as an alternative for the traditional phone service. Besides offering cheaper rates for long distance calls, VoIP freely supports a wide variety of services that users previously had to pay for, such as caller ID, voicemail, call waiting, call forwarding and call conferencing [21]. VoIP technology operates based on two types of standardized protocols: signaling and media protocols. Signaling protocols are responsible for call initiation, control, and termination, while media protocols are intended to carry voice data.

VoIP security has become one of the main concerns for both users and service providers since telephone conversations may carry sensitive and confidential information. Additionally, telephone services are used to verify the identity of the speaker as an authentication method. Therefore, whenever VoIP services are offered, they are expected to provide security services such as confidentiality, integrity, and authentication. However, providing secure VoIP services that are immune to the various types of attacks is a challenging issue, especially that many vulnerabilities related to the core IP network are inherited by VoIP. In addition, VoIP also suffers from signaling and media protocol specific vulnerabilities [8, 13, 31]. Furthermore, there is usually a trade-off between VoIP security and Quality of Service (QoS) [27]. The fact that the usage of VoIP over handheld devices has recently increased dramatically makes the process of implementing security mechanisms more challenging because of the power constraints of these handheld devices. We note here that securing the signaling of VoIP is outside the scope of this paper and the focus is on providing confidentiality of the media stream.

There is little work in the literature regarding efficient security mechanisms that meet both security requirements and processing overhead. It is known that conventional security mechanisms incur significant processing overhead and hence high energy consumption. This work proposes an energy-aware security mechanism for VoIP on handheld devices. This mechanism is based on a fact (demonstrated in this paper) that it is almost impossible for an attacker to predict the audio encoders' output frames of data (known as plaintext attacks) and the fact that these data frames contain much lower information density than that of textual data. The developed method reduces the processing time, which in turn reduces the consumed energy, while providing end-to-end security and maintaining good QoS from an end-to-end delay perspective. The proposed method is an enhanced security method over that proposed in [6]. The enhancement targets the required processing time to a level that is even lower than that of the Secure Real-time Transport Protocol (SRTP) yet without compromising end-to-end security.

The rest of the paper is organized as follows: Section 2 presents a literature review related to VoIP security in general and covers the work done in evaluating conventional and new security mechanisms. Section 3 introduces the proposed enhanced algorithm along with its analysis. In Section 4, the efficiency of the proposed mechanism is demonstrated experimentally. Finally, Section 5 concludes the paper.

2 Literature Review

The literature review covers two main categories: (1) VoIP signaling and media security, and (2) proposed security mechanisms:

2.1 VoIP Security

Since the core architecture of VoIP differs from that of the traditional Public Switched Telephone Network (PSTN), serious security issues are associated with VoIP and hence should be addressed [22]. The main challenge in VoIP security is to provide a service that possesses a level of security comparable to that of PSTN while maintaining an acceptable level of QoS and energy consumption.

Butcher et al. discuss general security issues related to VoIP and IP networks. They also discuss several attacks related to VoIP at the application level (attacks related to the Session Initiation Protocol (SIP)) suggesting different countermeasures to these attacks [4]. VoIP can also be protected by well-known security mechanisms such as IP Security (IPSec), Transport Layer Security (TLS), etc., with each having its own advantages and drawbacks. Barbieri et al. studied the impact of IPSec when used to secure VoIP. The results showed that the effective bandwidth is reduced by 50% in case of VoIP IPSec when compared to VoIP service alone [2]. Gupta et al. presented a structured discussion of VoIP security whereby they targeted three main aspects: media, signaling, and key derivation. They recommended that a replay-protected key exchange mechanism should be used along with SRTP [17]. Comparison of different security methods is also performed to investigate the impact on multimedia traffic. Hong et al. compared three main protocols: H.235, IPSec and SRTP. IPSec suffers from computational and bandwidth overhead over the other two protocols. SRTP seems to be the most suitable protocol due to the use of more advanced and modern cryptographic algorithms that take into account the QoS requirements of multimedia transmission [20]. Diab et al. conducted a comparison of different VPN security protocols that are used to protect VoIP data [11].

In order to answer the question of whether to use block or stream cipher to protect VoIP communications, Elbayoumy and Shepherd tried to compare the impact of AES cipher when it is applied both in block and stream mode to secure VoIP. Results showed that in terms of packet size, stream mode adds overhead to the packet less than block mode does. Results also showed that crypto engine performs better in case of stream cipher. Researchers concluded that subjective MOS and end-to-end delay measures gave better results in case of stream cipher compared to that of block cipher [12].

2.2 Evaluation of Conventional and New Security Algorithms

In order to evaluate the performance of SRTP and its effect on voice quality, Alexander et al. conducted an experiment to measure packet inter-arrival and jitter with and without security using a G.711 codec. The results showed that authentication is more time consuming than encryption [1]. New security algorithms based on selective encryption methodology as an alternative to conventional security mechanisms were also proposed in the literature. Servetti et al. proposed a new mechanism that partially encrypts the compressed bit stream at the output of a G.729 codec. The proposed method is based on the fact that the compressed bits have unequal perceptual importance. The proposed algorithm was subjected to both objective and subjective tests to prove its efficiency [29]. Choo et al. proposed a new lightweight mechanism to secure multimedia transmission and it was mainly proposed for video traffic. The algorithm involves two block transposition operations along with a single XOR operation on each video frame. Experiments showed that the new proposed algorithm is three times faster than applying the Advanced Encryption Standard (AES) on video data. It was also shown that Secure Real Time Media Transmission (SRMT) is better than previously proposed mechanisms in terms of security and QoS [9].

In order to reduce the complexity of encrypting a voice stream to fulfill the power constraints of handheld devices, a new method based on selective encryption for Moving Picture Experts Group (MPEG) voice streams was proposed by Servetti et al. The method exploits the fact that a voice stream can be divided into perceptual and non-perceptual parts and it achieves security by only encrypting the perceptual bits [30]. Wu et al. applied syntax-aware selective encryption that takes into account communication and transmission constraints. The location of the encryption process within the bit stream is also discussed [33]. Xie et al. introduced a new method to encrypt the compressed bit stream that represents the output of entropy encoders. It is stated that because the resulting bit stream at the output of the encoder has significant randomness, it is not necessary to perform heavyweight cryptographic techniques, and hence inserting a simple randomness operation in the stream is sufficient [32]. Han et al. proposed a new encryption method for multimedia content on handheld devices and it works by alternating between AES in block mode and RC4 in stream mode. The proposed method was evaluated using desktop computers and MPEG Layer III (MP3) audio files [19]. Abou Charanek et al. proposed a new method for encrypting voice traffic based on selective encryption called Energy Efficient Voice over IP Privacy (E^2VoIP^2) [6]. The study showed that encrypting the voice traffic with conventional algorithms consumes a significant amount of energy in addition to the introduced delays, specifically for handheld devices. Compared to SRTP, E^2VoIP^2 is more efficient in terms of CPU cycles and processing time when it is implemented on HP iPAQ handheld devices. The proposed mechanism was based on mixing a block cipher with a stream cipher by simply applying an AES block cipher on the first packet which is padded with a random number within segmented groups, and performing XOR operation on the remaining packets within the same group using the corresponding random

number.

SRTP is one of the most popular security mechanisms used to secure media streams in VoIP applications. SRTP has a very low overhead and it is the secure version of the traditional Real Time Protocol/ Real-time Transport Control Protocol (RTP/RTCP) which is mainly used for real-time transmission of multimedia over IP. SRTP provides confidentiality, integrity, authentication, and replay protection for RTP and RTCP traffic. In SRTP, AES in counter or f8 mode and HMAC-SHA-1 are the predefined algorithms for encryption and authentication, respectively [3].

Although there is significant work in the literature that aims at developing an efficient VoIP security solution, only a few of them took into consideration the limited energy capabilities of handheld devices. Most of the proposed selective encryption methods are codec-dependent and are rarely tested and assessed on handheld devices [16]. In the next sections, we describe our proposed algorithm (SecVoIP) and show how it achieves a good balance between energy consumption and security without affecting the quality of the exchanged voice. We will highlight the enhancement of SecVoIP as compared to the recent algorithm presented in [6], namely, E^2VoIP^2. The first improvement is related to the generation of the random number that was added to the first packet of each group in E^2VoIP^2. This random number was used to encrypt/decrypt the remaining packets of the group. Therefore, any loss of the first packet of a group leads to discarding the remaining packets in the group. We overcome this weakness by eliminating the padding process of any additional data to the original packet. The generation of the random number that is used to decrypt the packets of the group can still be performed even if there is a packet loss. Moreover, avoiding the padding process solves major drawbacks in E^2VoIP^2 such as additional bandwidth and time consumption, which resulted from adding such extra information to the packet. Moreover, the comparative assessment was performed on a modern platform consisting of a Samsung Nexus S smartphone running the Android 2.3 operating system.

3 SecVoIP Design and Analysis

As mentioned previously, SRTP is the most widely used standardized protocol to secure VoIP communications. The global trend to develop more energy-efficient mechanisms for battery-powered handheld devices motivated the design and implementation of an energy-efficient security mechanism that outperforms SRTP in terms of processing time and energy consumption without compromising the security of the data. As in [6], the attacker model incorporates packet sniffing, replaying, dropping, and reordering capabilities. Based on experimental and analytical analysis presented in [6] and further validation in this work, it is demonstrated that such an attacker is not capable of performing known plain text attacks even with direct access to the raw signal.

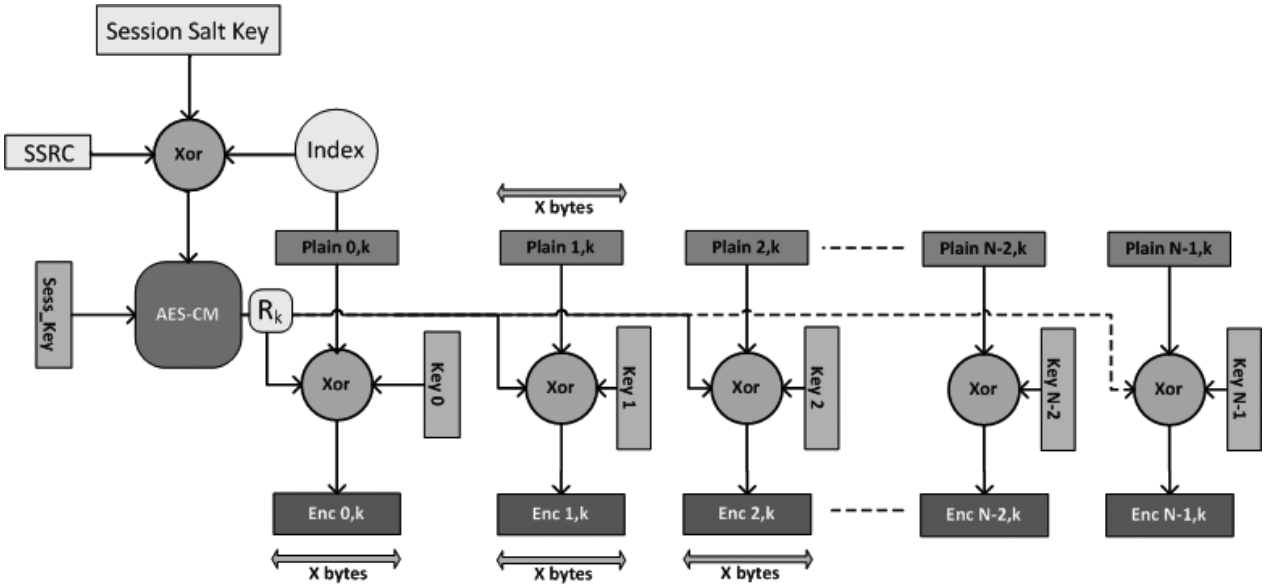


Figure 1: Design of SecVoIP encryption algorithm

3.1 SecVoIP Design

SecVoIP combines the mechanism of E^2VoIP^2 in encrypting voice packets and the mechanism of SRTP in generating the key stream. As shown in Figure 1, voice packets are divided into groups of N packets each. For the first packet of each group, AES in counter mode is applied to generate a pseudorandom stream involving attributes similar to those used in SRTP. Afterwards, every packet in the group is encrypted by XOR-ing the triplet: plaintext, random number, and the predefined key at every packet position. Let X be the size of the plaintext in bytes which is

defined by the encoder in use. Considering a group k of N packets as shown in Figure 1, every plain packet ($Plain_{x,k}$) in the group is encrypted based on Equations (1) and (2).

$$Enc_{x,k} = Plain_{x,k} \oplus Key_x \oplus R_k \quad (1)$$

$$Let\, K_{xk} = Key_x \oplus R_k \Rightarrow Enc_{x,k} = Plain_{x,k} \oplus K_{xk} \quad (2)$$

The pseudorandom string R_k of X bytes used in Equation (1) is distinct for every group. One major difference between SecVoIP and SRTP is that instead of generating a pseudorandom string for every packet being transmitted and received, SecVoIP does so only once at the beginning of every group of N packets. Therefore, R_k for a certain group is calculated over an initial value (IV_k) using:

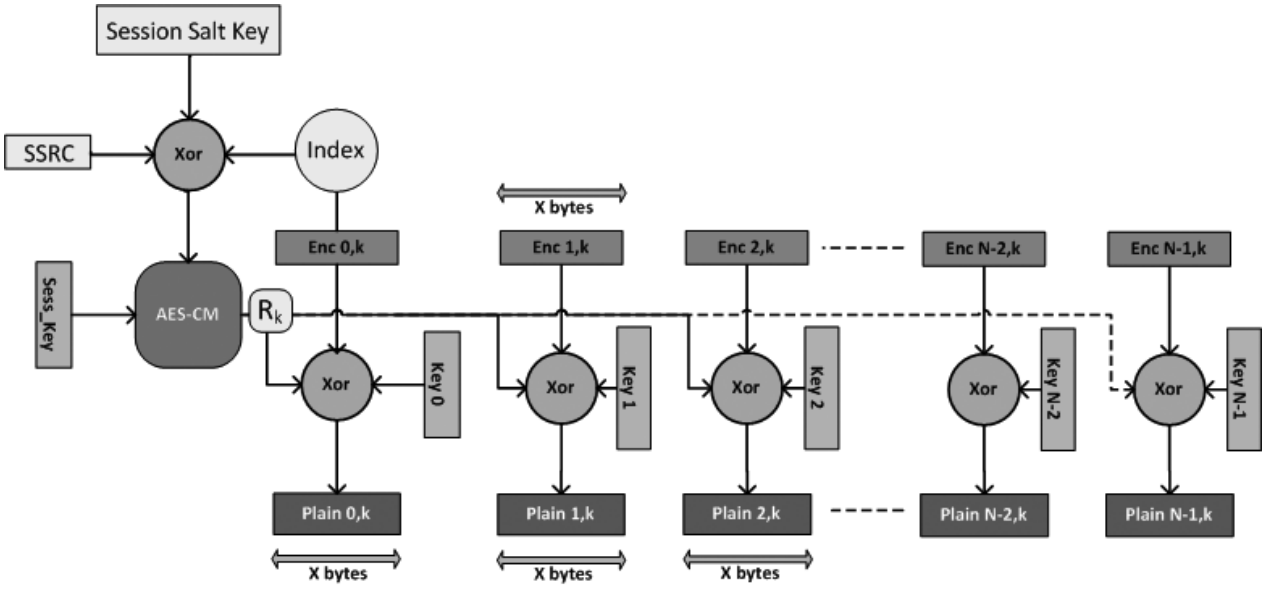


Figure 2: Design of SecVoIP decryption algorithm

$$IV_k = (SSRC) \oplus (Session\_Salt\_Key) \oplus (index_{k,0}) \quad (3)$$

Where SSRC is a 32 bit Synchronization Source Identifier that is chosen randomly and Session Salt Key is a random number used to defeat pre-calculation attacks, whereas $index_{k,0}$ is formed based on the Sequence Number (SEQ) of the first packet in group *K* and current Roll Over Counter ROC as defined in [3]:

$$Index = 2^{16} \times ROC \oplus SEQ$$

As soon as IV_k is calculated, it is fed to AES in counter mode to generate R_k. The same procedure is repeated for the next group (K+1) of packets and a distinct R_{k+1} is generated. This process continues as long as voice packets are produced by the encoder.

In addition to session key and session salt keys, which are used by AES to generate the random stream that will be used to encrypt the packets in the group, N predefined keys of size X bytes need to be shared between the sender and the receiver as in E^2VoIP^2. However, E^2VoIP^2 did not suggest any special mechanism to exchange these keys. Therefore, instead of using conventional key exchange mechanisms which may be considered costly, we suggest using SRTP's key generation mechanism [3]. From a single exchanged master key and master salt, we can generate all necessary keys by assigning different label values for each key at a certain position. Hence, both sides can agree on a key derivation rate at which all keys are refreshed, thus increasing the security of the method. In either case, key derivation must be performed once at the start of the conversation such that sufficient keys are supplied to the encryption and decryption modules. Any mechanism can be used to exchange the required master key, master salt, and key derivation rate. However, there are various key exchange mechanisms proposed in the literature in the context of VoIP security. Among these mechanisms are SDES, MIKEY, ZRTP, DTLS-SRTP and others [1]. MIKEY is the most widely used mechanism for SRTP. Any of these key exchange mechanisms can be used to share the required keys between parties. Gupta et al. present a detailed security analysis for each of these key exchange mechanisms along with some security considerations in applying each of them [17]. The exchanged keys should be kept secret and stored by the caller and the callee within a cryptographic context along with other transformation related parameters. The additional overhead caused by the generation of additional keys is analyzed later in this section. Note that the attributes involved in the generation of the random number R_k are all known to the receiver except the ROC which is maintained by the receiver using the mechanism proposed in [3] or any other efficient mechanism. This fact eliminates the need to pad the random number to the first packet of each group, giving SecVoIP a major advantage over E^2VoIP^2.

Assuming that the number of packets per group (N) is known to the receiver, decryption is performed in the same way as encryption, as shown in Figure 2, and hence only the encryption process is analyzed in the next section.

3.2 SecVoIP Analysis

In the following section, we analyze the proposed algorithm from a networking and a security perspective.

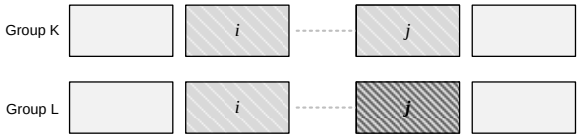
3.2.1 Bandwidth Overhead and Packet Loss

Assuming that all the previously-mentioned algorithms are using RTP as the media-carrying protocol, the ability to generate the random number at the receiver side without the need to transmit additional data allowed the bandwidth consumption to be equal to that of SRTP.

From a packet loss perspective, and since all the attributes involved in generating the group-specific random number can be generated by the receiver, even if the first packet of the group is lost, synchronization between sender and receiver can be maintained which is not the case in E^2VoIP^2. Packet loss effect in SecVoIP is similar to that in SRTP. So, the same packet loss concealment methods used in conjunction with SRTP can be used in SecVoIP.

3.2.2 Security Concerns

A security analysis is performed in [6] in order to ensure the security of the proposed mechanism. It is shown that E^2VoIP^2 is theoretically immune against different scenarios of known ciphertext and known plaintext/ciphertext attacks. Since SecVoIP is designed based on the basic principles of both SRTP and E^2VoIP^2, it inherits the security properties of both. The same security concerns of SRTP mentioned in [3] are also applicable to SecVoIP. In addition to the scenarios mentioned in [6], there is a specific scenario to which SecVoIP and E^2VoIP^2 are vulnerable. The success of this attack highly depends on the ability of the attacker to deduce some of the plaintext packets in the voice stream. If the attacker is able to know the plaintext packets at positions i and j in a certain group K and a plaintext packet at position i within a different group L, the attacker can deduce the plaintext packet at position J in group L.



From Equations (1) and (2):

$$Enc_{i,k} \oplus Enc_{j,k} =$$
$$(Plain_{i,k} \oplus R_k \oplus Key_i) \oplus (Plain_{j,k} \oplus R_k \oplus Key_j) = C$$
$$\Rightarrow Plain_{i,k} \oplus Key_i \oplus Plain_{j,k} \oplus Key_j = Key_i \oplus Key_j = C$$

$$Enc_{i,L} \oplus C =$$
$$(Plain_{i,L} \oplus R_L \oplus Key_i) \oplus (Key_i \oplus Key_j) = W$$
$$\Rightarrow (Plain_{i,L} \oplus R_L \oplus Key_j) = R_L \oplus Key_j = W$$

$$Enc_{j,L} \oplus W =$$
$$(Plain_{j,L} \oplus R_L \oplus Key_j) \oplus (R_L \oplus Key_j) = Plain_{j,L}$$

In general, this type of attack becomes highly sophisticated if the nature of the plaintext prevents the attacker from predicting or estimating some packets. In previous work, the authors presented exhaustive experimental results to demonstrate the fact that it is not possible to deduce voice plaintext at encoder output, and hence it is not feasible to perform known plaintext/ciphertext attacks [6]. In these experiments, specialized hardware and software systems were used in order to investigate various realistic scenarios with variable environmental characteristics. The main purpose of the experiments which are further detailed in [7] was to measure the similarity between two instantaneously recorded sound files in an acoustically controlled environment and under different scenarios with respect to the position of the microphones.

The captured voice by the two microphones of the attacker and the victim were encoded with G.729 codec. Based on the fact that the encoder produces 10-byte frames, and instead of comparing the whole two files together, the two following methods are used to measure the similarity:

- One of the two binary files is broken into frames of 80 bits, and for each frame a sliding window with a size of 80 bits is shifted bit by bit through the other file. For every bit shift, a binary similarity coefficient is measured between the two 80-bit strings.
- The two binary files are broken into frames of 80 bits. Each frame is compared to all other frames in the second file.

Table 1 shows the average similarity measure values for both techniques and for all test cases specified in [6].

Table 1: Average percentage of similar bits

Technique	Average percentage of similar bits
Bit shift	50.3%
Frame shift	53%

We also calculated the average of the maximum similarity values for each of the cases and the results were very close to that of two randomly generated files as shown in Table 2.

Table 2: Average of maximum similarity measure

Technique	Recorded Files	Random Files
Bit shift	73.9%	73.75%
Frame shift	70%	67.5%

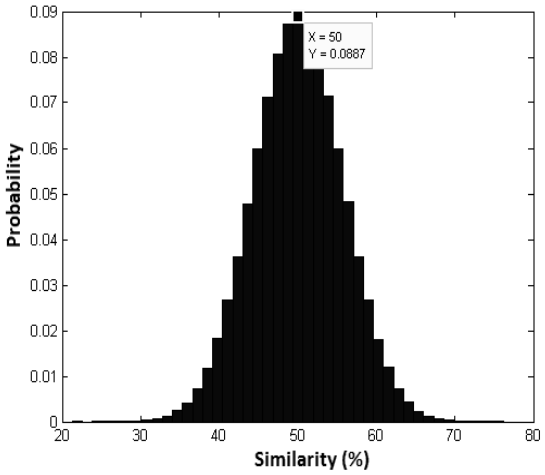


Figure 3: Similarity measures distribution of two random files

In addition to the statistical results and studies presented in [6], further analyses are performed based on the outcome of the previous experiment in this paper. The distribution plots for various scenarios are presented and analyzed. Because the distributions of similarity measures for all scenarios are almost identical to each other, we only include the normalized distribution plots of the similarity measures for three different scenarios including the case in which the victim's and attacker's microphones are at the same distance with angle 0, which is considered a worst case scenario. These distributions are depicted in Figure 3 and Figure 4. Based on the values and the figures presented, we can conclude that the similarity measures behave like a random variable with binomial distribution representing the discrete probability distribution of the number of matches in successive 80 independent Bernoulli trials with a probability of 0.5 for both match and mismatch. Therefore, this demonstrates that the two output files look almost like two randomly generated files.

In order to confirm that the results obtained are codec-independent, a similar procedure was performed using the "Speex" codec. Similar outcomes were obtained. It is expected that all CELP based encoders would provide similar results.

We have seen in [6] that introducing a natural silence of 10ms at the beginning of the conversation breaks the similarity. To study the worst case scenario, another test

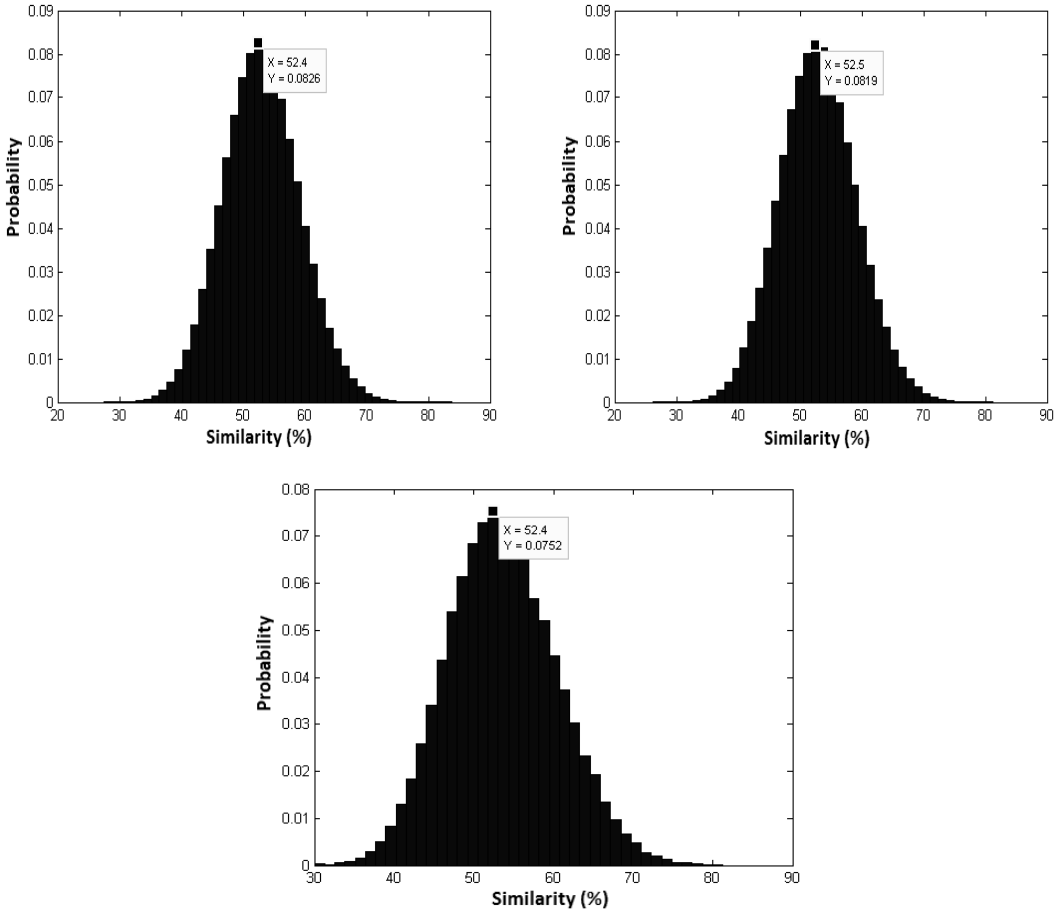


Figure 4: Similarity measures distribution of three different scenarios

was conducted by adding a frame of 160 bytes of 0s at the beginning of the recorded file and then measuring the similarity between the original and the modified one. The added bytes represent a single 10ms frame of speech at the beginning of the G.729 encoded file. After applying the same comparison methods, the results showed that adding just a 10ms of 0 bits to a file containing human speech is enough for the encoder to break the similarity and to reduce the maximum similarity value from 100% to 75.8% on average. A string of random bits instead of a string of 0s was also added at the beginning of the file and the maximum similarity was further reduced to 66.3%. It is also observed that for computer-generated sound, the longer the added string at the beginning of the file, the higher the reduction in the maximum similarity. It is believed that this is related to the fact that differential encoders reset variables in order to reduce the propagation of error after a given duration causing the 100% similarity for the frames occurring after this reset. This outcome is very important because it clearly demonstrates how a very simple modification in the original speech propagates and affects many subsequent frames. Therefore, if one of these 10ms frames exists at any location in the streams recorded by the two microphones, it would be practically impossible for an attacker to reproduce the exact bit stream.

Therefore, and quoting from [3], "It is difficult for an adversary to acquire the RTP plaintext data, since for many codecs, an adversary that does not know the input signal cannot manipulate the output signal in a controlled way. In many cases it may be difficult for the adversary to determine the actual value of the plaintext.", and from the results in [6] and the results presented in this paper, a strong case is made that it is not feasible for an attacker to perform known plaintext/ciphertext attacks due to the stochastic nature of the output of the codec. Nevertheless, the only threat model that affects the proposed algorithm from a confidentiality perspective is the ability of the attacker to successfully perform an attack by using the same exact input as the victim's. However, this requires direct physical access to the victim's device in order to record the conversation, which is assumed beyond the attacker model considered in this paper.

3.2.3 Group Size and Key Management Analysis

The lower bound for N, size of group used, is defined such that the ratio of time taken to encrypt N packets using SecVoIP to the time taken to encrypt N packets using SRTP is lower than a certain threshold. Let AES[S] be the time taken to encrypt a data block of size S bytes with AES and XOR[S] be the time taken to XOR two chains of size S bytes. The times needed to encrypt a group of N packets using SecVoIP ($\tau_{SecVoIP}$) and SRTP (τ_{SRTP}) are:

$$\tau_{SecVoIP} = \frac{AES[X+16-(X \bmod 16)] + 2N \times XOR(X)}{N} \quad (4)$$

$$\tau_{SRTP} = N \times \left[\frac{AES[X+16-(X \bmod 16)]}{N} + \frac{XOR(X)}{N} \right] \quad (5)$$

It is obvious that the size of the packet on which AES is performed is smaller than that in the previous E^2VoIP^2 since there is no additional padding of bytes. Let γ be the average time taken to XOR a packet of X bytes over the time taken to encrypt a packet with SRTP. The ratio of time taken by SecVoIP to the time taken by SRTP is given by:

$$\Omega = \frac{\tau_{SecVoIP}}{\tau_{SRTP}} = \frac{AES[X+16-(X \bmod 16)] + 2N \times XOR(X)}{N \times AES[X+16-(X \bmod 16)] + N \times XOR(X)} \quad (6)$$

$$\Rightarrow \frac{AES[X+16-(X \bmod 16)] + 2N \times XOR(X)}{N \times AES[X+16-(X \bmod 16)] + N \times XOR(X)} < \rho$$

$$\Rightarrow \frac{[AES(Z) + XOR(X)]}{\rho \times N \times [AES(Z) + XOR(X)]} + \frac{(2N-1) \times XOR(X)}{\rho \times N \times [AES(Z) + XOR(X)]} < 1$$

$$\Rightarrow \frac{1}{\rho \times N} + \frac{(2N-1) \times \gamma}{\rho \times N} < 1$$

$$\Rightarrow 1 + (2N-1) \times \gamma < \rho \times N$$

$$\Rightarrow N > \frac{1-\gamma}{\rho - 2\gamma} \quad (7)$$

The condition given in Equation (7) is experimentally validated in the subsequent sections. Although there is no certain upper bound for N, as we will see later, increasing N increases the efficiency of the proposed algorithm at the cost of generating the additional number of predefined keys needed for encryption. Comparing the time needed by SRTP and the time needed by SecVoIP in generating the required keys we can compute the additional key generation overhead incurred by SecVoIP.

Assuming that both the sender and the receiver share for every session a master key and a master salt, two 128-bit keys are generated: session key and session salt key, which are refreshed every ***R*** packets. SecVoIP requires ***N*** keys of X bytes in addition to these two session keys. These additional keys may also be refreshed every ***R*** packets. Let P_{SRTP} and $P_{SecVoIP}$ represent the time needed to generate the keys in SRTP and SecVoIP, respectively, for every ***R*** packet:

$$P_{SRTP} = 2 \times AES[16] \quad (8)$$

$$P_{SecVoIP} = 2 \times AES[16] + N \times AES[X-(X \bmod 16)] \quad (9)$$

Comparing Equations (8) and (9), the larger the number of packets in a group (N), the larger the key generation overhead of SecVoIP as compared to SRTP. This one time additional overhead is overcome by the gain acquired in the encryption process of every packet. The total time taken to encrypt R packets can be given by:

$$\tau'_{SRTP} \approx R \times AES[X+16-(X \bmod 16)] + R \times XOR(X) \quad (10)$$

$$\tau'_{SecVoIP} \approx Ceil\left(\frac{R}{N}\right) \times AES[X+16-(X \bmod 16)] + 2R \times XOR(X) \quad (11)$$

For R=N, there will be (N) AES operations in SRTP

and (N+1) AES operations in SecVoIP in total for every R packet. In this case, SRTP outperforms SecVoIP by one less AES operation which clearly indicates that R should be greater than N. In general, assuming that R is larger than N and that at least R packets are exchanged, the relation between ***R*** and ***N*** such that SecVoIP outperforms SRTP, can be derived as follows:

$$G = \tau'_{SRTP} - \tau'_{SecVoIP} - P_{SecVoIP} + P_{SRTP} > 0$$

$$\Rightarrow R \times AES[Z] + R \times XOR(X) - Ceil\left(\frac{R}{N}\right) \times AES[Z]$$

$$- 2R \times XOR(X) - N \times AES[Z] > 0$$

For simplicity, assume that, $XOR(X) = k \times AES[Z]$, where k < 1, therefore:

$$\Rightarrow R(1+k) - Ceil\left(\frac{R}{N}\right) - 2RK > N \quad (12)$$

Given that R and N are integer values, R can be written as:

$$R = qN + r \quad (13)$$

Substituting Equation (13) in (12):

$$\Rightarrow q > \frac{N + kr - r}{N - kN - 1}$$

Therefore, for $q = 1$, which represents the worst case scenario in terms of key generation cost:

$$\Rightarrow N + kr - r < N - kN - 1 \Rightarrow r > \frac{kN + 1}{1 - k} \quad (14)$$

$$\Rightarrow R > N + \frac{kN + 1}{1 - k} \quad (15)$$

Since XOR operation is negligible compared to AES, without loss of generality, Equation (14) after substituting *k=0* becomes:

$$\Rightarrow r > 1$$

The next possible positive integer for r is 2. As a result, in order to have a positive gain G, R should be greater than (N + 2). The previous analysis shows that increasing the number of packets per group increases the number of required predefined keys which in turn increases the key management overhead. However, this overhead is compensated for by the encryption process as long as the condition of Equation (14) is satisfied. When the rate of refreshing keys R=0, the keys are not refreshed. In this case, for SecVoIP to outperform SRTP, the number of exchanged packets should exceed N by 2, which is considered as a very relaxed condition.

The next section includes the implementation of the proposed algorithm along with experimental results that demonstrate the efficiency of the algorithm in terms of processing time and energy.

4 Implementation and Experimental Results

The prototype implementation of SecVoIP was done using Csipsimple [5], which relies on PJSIP [25], installed on a Samsung Nexus S smartphone running the Android 2.3 operating system. SRTP is used as the benchmark algorithm. PJSIP's SRTP implementation is used to calculate the processing time of SRTP encryption. In order to study the impact of having different number of packets per group on the efficiency of the algorithm, two group sizes, N=5 and N=15, are used. The analytical study previously presented regarding the group size is experimentally validated in this section.

First, the energy efficiency of SecVoIP over SRTP is demonstrated in terms of time consumption and number of CPU cycles which are directly related to the consumed energy. The outcomes are further validated through direct energy measurements performed on the Android phone. Note that the presented experimental results do not include the initial key generation cost, which was analyzed in the previous section. The cost of implementing several AES operations at the beginning of the session to generate the keys is very small compared to the cost of implementing AES to encrypt the large number of voice packets generated throughout the session.

4.1 PJSIP

PJSIP softphone is an open source application written in the C language and provides basic and advanced VoIP features [24]. PJMEDIA is a fully featured stack that controls the media component of PSJIP. PJMEDIA-CODEC contains a wide variety of well-known voice codecs such as G.711 (μ-law and a-law), G.722, GSM, and others that are integrated into PJMEDIA framework. Additionally, PJSIP provides SRTP functionality through the libsrtp() library. The SRTP module is plugged in between the stream block and the transport block. Csipsimple is a project relying on PJSIP to provide native SIP functionality for Android devices [5].

4.2 Time Consumption: Results and Analysis

Creighton et al. showed how energy consumption is directly related to the time taken to encrypt data using a certain security algorithm [18]. Similarly, Diaa et al. show how the packet size and the time consumed to encrypt this packet affect throughput and consumed energy. For the same packet size, higher encryption time results in lower throughput and higher energy consumption [10].

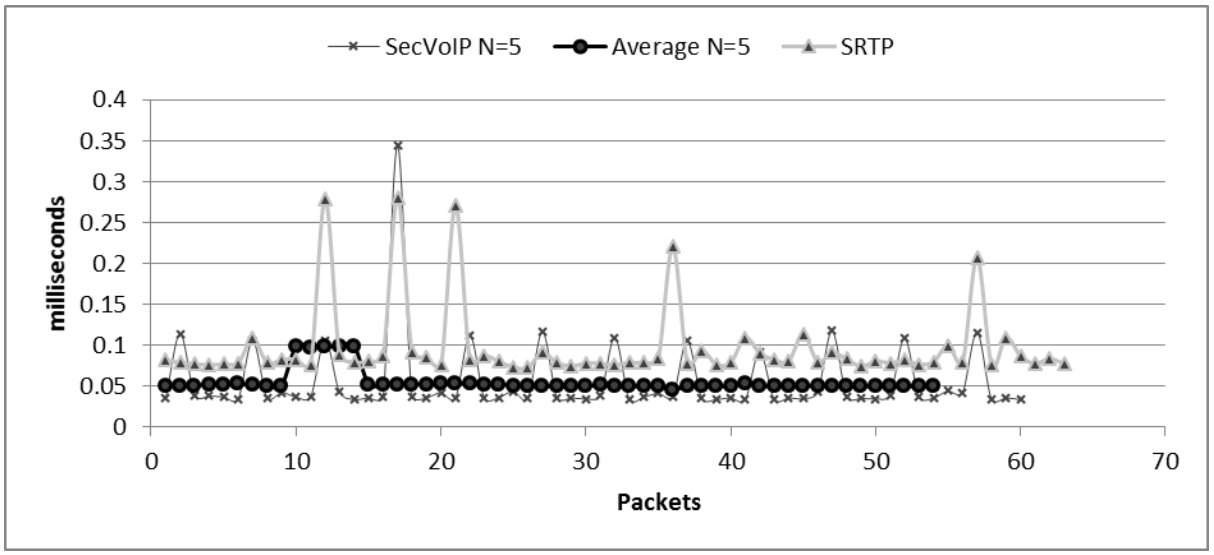


Figure 5: GSM encryption time in milliseconds for N=5

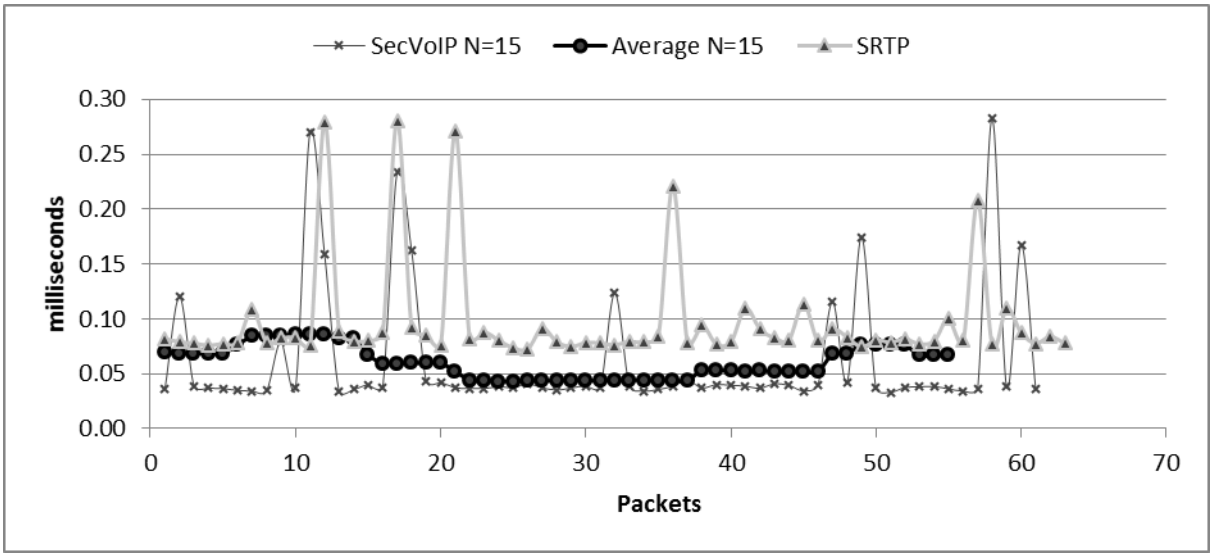


Figure 6: GSM encryption time in milliseconds for N=15

In order to calculate the time taken to encrypt a packet with SecVoIP and SRTP, timing markers were introduced into the code in order to calculate the processing time required by the encryption. For similar computations between SRTP and SecVoIP, i.e. AES counter mode, special care was taken to use identical functions in both algorithms such that we obtain as fair a comparison as possible. During the experiments, two types of codecs, G.722 and GSM, were used to investigate the relationship between the processing time and the payload size. The G.722 codec operates at 64 kbps and produces a payload of 160 bytes every 20ms, whereas the GSM codec produces 33 bytes of 20ms voice payload. For each codec, two group sizes were considered: N=5 and N=15. Because decryption in our algorithm for all possible N values works exactly the same way as encryption, only results for the encryption part are presented. It is worth mentioning that during the experiments, random spikes of up to 2 ms in the processing times appeared. These spikes are operating system related and are not specific to SecVoIP or SRTP. However, these spikes are all included in our figures and measurements.

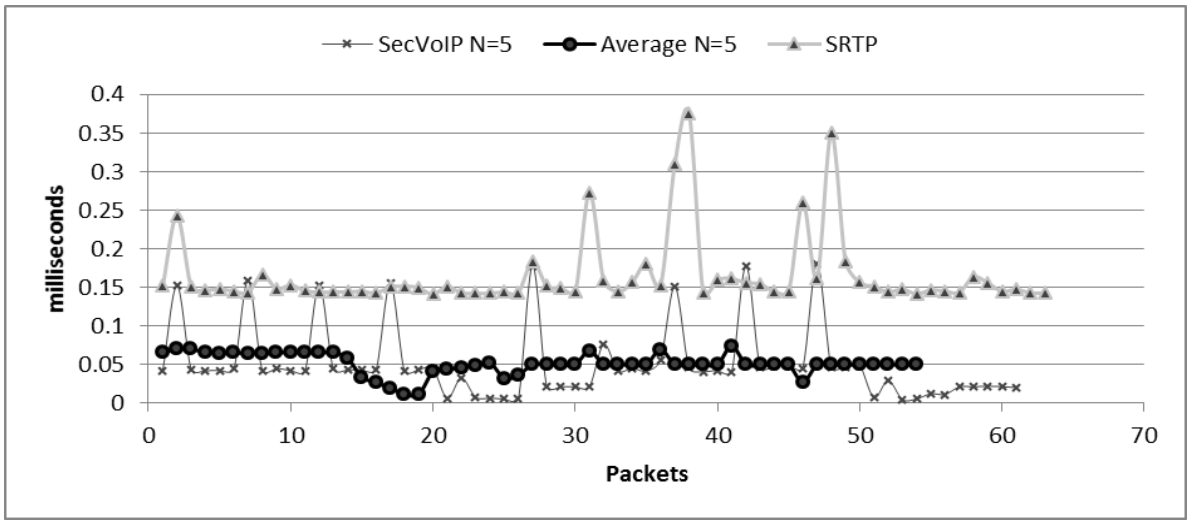


Figure 7: G.722 encryption time in milliseconds for N=5

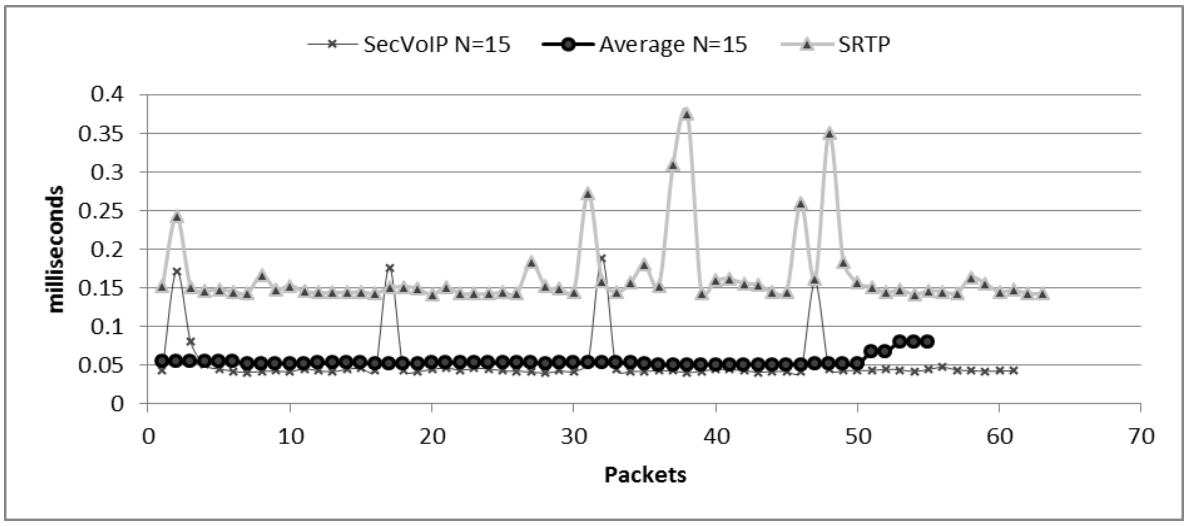


Figure 8: G.722 encryption time in milliseconds for N=15

Figure 5 and Figure 6 show the average processing time of SecVoIP for N=5 and N=15, respectively, for GSM-encoded packets. We can clearly notice the additional overhead added to the first packet of each group resulting from the generation of the pseudorandom sequence of X bytes. Results show that, on average, for 10 minutes of voice conversation, SecVoIP has 53% and 58% less operation time than that of SRTP for N=5 and N=15, respectively. Because the way SecVoIP encrypts the first packet of every group is similar to how SRTP encrypts each and every packet, they almost have identical processing times.

The same was done for packets encoded with G.722 codec. Figure 7 and Figure 8 represent the average processing time needed to encrypt G.722 encoded packets with SecVoIP and SRTP for N=5 and N=15, respectively. On average, SecVoIP is 58% faster than SRTP for N=5 and 83% faster for N=15.

Table 3 shows the smallest possible value for N obtained from the condition in Equation (7) for which γ is extracted from the plots. It illustrates the existing inverse relationship between N and Ω such that for a certain packet size, the higher the value of N, the lower the Ω ratio.

Based on the results presented above, it can be concluded that for the G.722 codec, SecVoIP outperforms SRTP in terms of processing time for the two group sizes, whereas for the GSM codec, the group size should be greater than 8.2. As analyzed previously, increasing the group size improves the efficiency over SRTP even further.

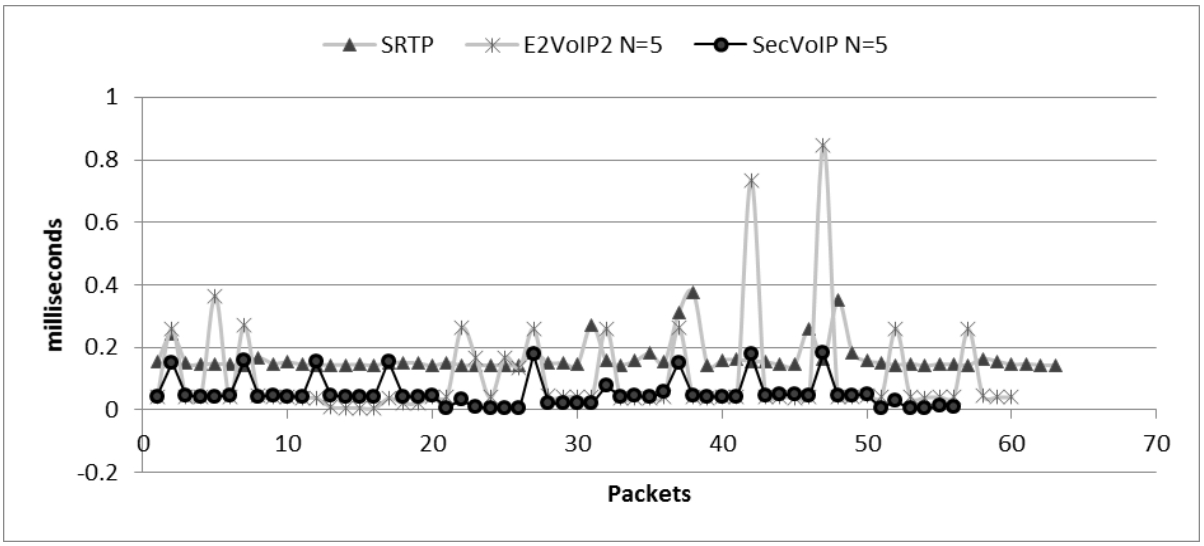


Figure 9: SecVoIP versus SRTP and E^2VoIP^2 for G.711 codec and N=5

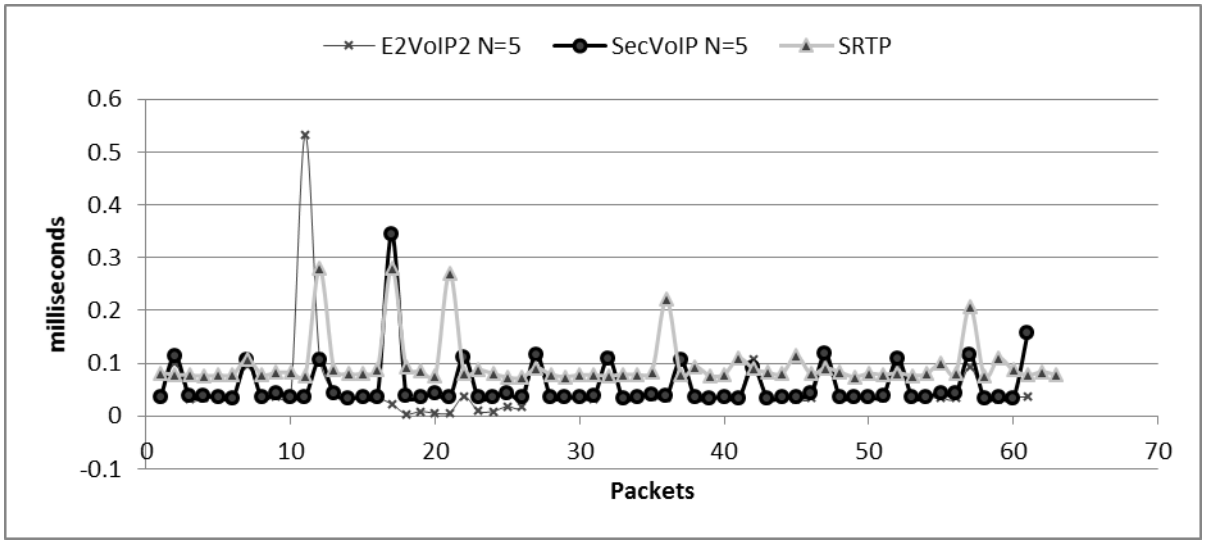


Figure 10: SecVoIP versus E^2VoIP^2 and SRTP for GSM codec and N=5

Table 3: Results in terms of different N Values

Codec	Packet Size "X"	N	Ω(N,X)	ρ(X)	γ(X)	η = (1-γ)/(ρ-2γ)
GSM	33	5	0.46	0.46	0.18	8.2
		15	0.42			
G.722	160	5	0.42	0.53	0.18	4.8
		15	0.208			

In order to compare the experimental results with those of the previous E^2VoIP^2 algorithm presented in [6], E^2VoIP^2 was also implemented on the same Android device. As a result, for G.711 codec, a significant decrease in the E^2VoIP^2 processing time in encrypting the first packet of each group is observed as depicted in Figure 9.

In E^2VoIP^2, AES was applied on a packet size of 320 bytes, which translated into twenty AES operations. In SecVoIP, on the other hand, AES-CM is used to generate a random sequence of 160 bytes, which requires 10 AES operations. In other words, the time needed to encrypt the first packet of each group in SecVoIP is half of that needed in E^2VoIP^2. However, the total operational time is not reduced exactly by 50% due to the additional XORs required by SecVoIP. On the other hand, for the GSM codec, and as is clear from Figure 10, the difference is hard to notice because in E^2VoIP^2 AES is applied on packets of size 64 bytes after padding, while in SecVoIP it is applied on packets of size 48 bytes.

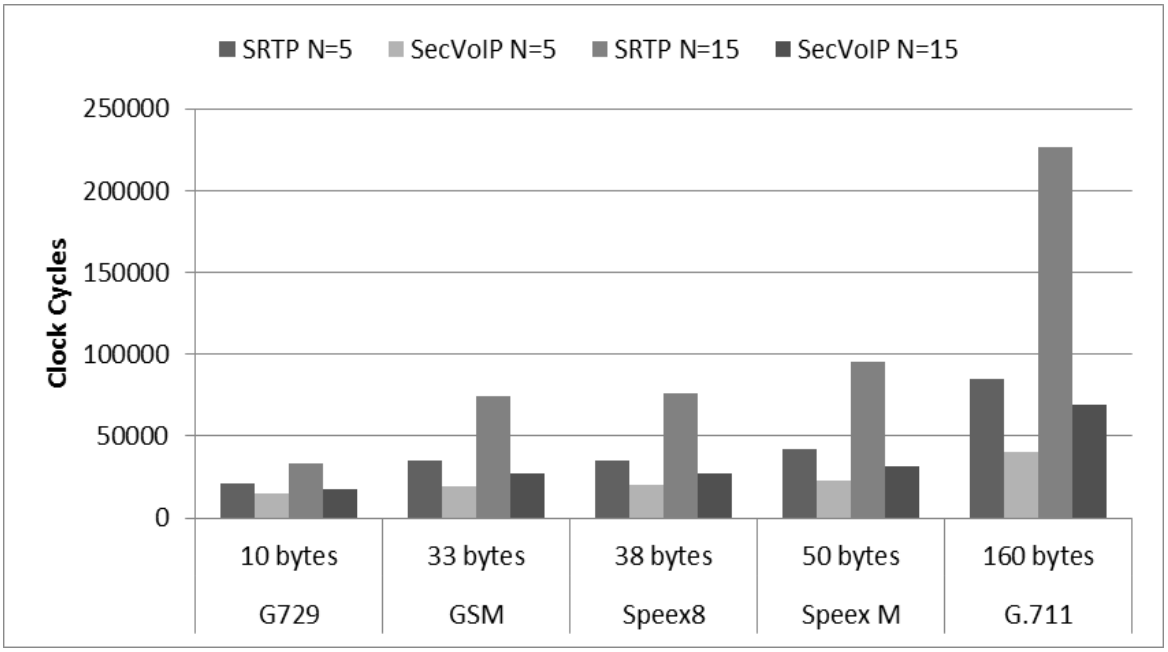


Figure 11: Number of CPU cycles for N=5 and N=15

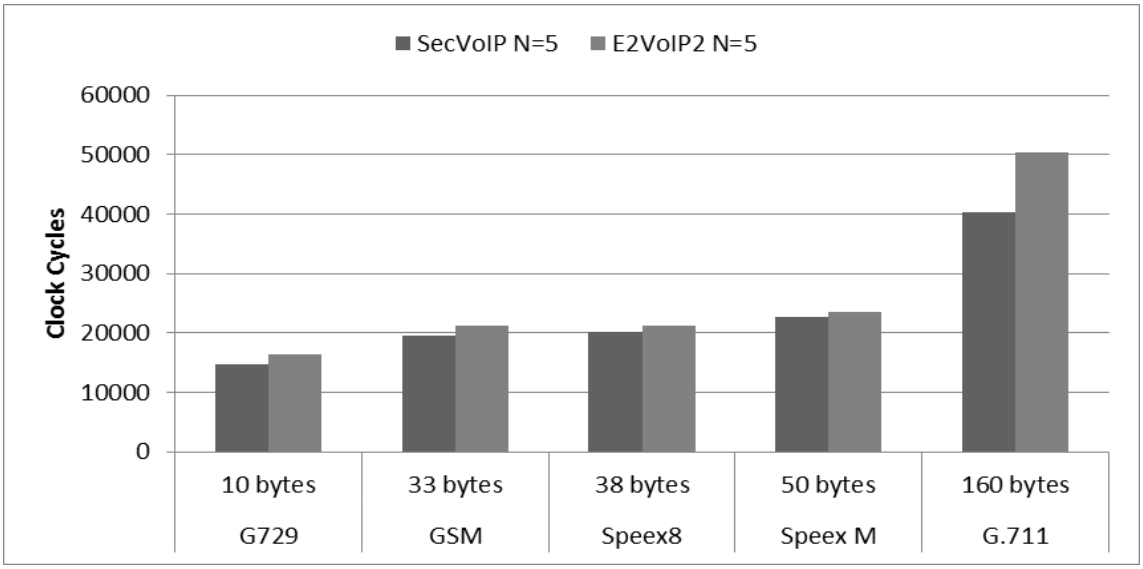


Figure 12: Number of CPU cycles for E^2VoIP^2 versus SecVoIP

4.3 CPU Cycles: Results and Analysis

Ruangchaijatupon et al. calculated the energy consumed by a cryptographic algorithm based on the number of the required CPU cycles. Researchers indicated that the consumed energy from any process is directly related to the CPU cycles consumed by the instructions of the corresponding process [23, 28].

To demonstrate the efficiency of our algorithm in terms of CPU cycles, we used PTLsim as a test bed in order to calculate the number of CPU cycles consumed during encryption. PTLsim is an accurate x86 microprocessor simulator, which is used to simulate x86 and x86-64 instructions [26]. An application was written in C++ to simulate SRTP and SecVoIP encryption. Various codec types are selected in order to cover a wide range of packet sizes. As in the previous section, two values for N were selected: 5 and 15.

Figure 11 depicts the number of CPU cycles required for different packet and group sizes and shows significant improvement of SecVoIP over SRTP. It also shows that the amount of savings is in a direct relation with the number of packets per group and packet size.

The ratios shown in Figure 11 differ from those presented in the previous figures mainly due to the dissimilarity of the processor model used (ARM versus X86). For the same reason stated previously, the savings in SecVoIP as compared to E^2VoIP^2 is of a greater significance for 160-byte data than the others, as shown in Figure 12. Nevertheless, both mechanisms show a great enhancement over SRTP.

4.3 Power Measurements

In order to demonstrate the energy efficiency of SecVoIP by means of real physical measurements, the same application developed in the previous section was compiled and deployed on the Samsung Nexus S Android phone.

A SIP server was used to create SIP accounts. The Android phone was assigned a SIP account, and a softphone was assigned another account. The goal was to measure the energy consumption on the Android phone when calls to the softphone, running on a desktop computer, were being made.

Three applications were installed on the Android phone; E^2VoIP^2, SecVoIP, and the original CSIPSimple application as downloaded from the PJSIP website. Experimental calls were made to measure the energy consumption when deploying E^2VoIP^2, SecVoIP and CSIPSimple with SRTP encryption enabled. During these calls the same audio snippet was played repeatedly.

Table 4: Energy measurements (Joules) for all cases

Codec	Packet Size (Bytes)	Algorithm	Energy (J)
GSM	33	SRTP	65
		E^2VoIP^2, N=5	52
		E^2VoIP^2, N=15	46
		SecVoIP, N=5	47
		SecVoIP, N=15	43
G.722	160	SRTP	171
		E^2VoIP^2, N=5	124
		E2VoIP2, N=15	119
		SecVoIP, N=5	123
		SecVoIP, N=15	103

In addition, the Android phone and the PC that has the softphone running were always kept at the same distance from each other and from the router. This was all done to ensure that the measurements were being recorded in a controlled environment. For every application, five 10-minute calls were made in a random order, and energy measurements were recorded using the PowerTutor application [15]. The results for all cases were then averaged and are displayed in Table 4.

The results clearly indicate that SecVoIP outperforms both E^2VoIP^2 and SRTP in terms of the consumed energy. For the case of GSM with N = 5, the percent improvement in energy consumption between SRTP encryption and SecVoIP is 28%, and is 34% for N = 15. For the case of G.722 and N = 5, the percent improvement is 28%, and increases to 40% for N = 15.

In order to translate these results into battery lifetime, we first measured the energy consumption of the Samsung Nexus S phone when in standby mode with the WiFi and 3G radios ON and the screen OFF. Table 5 lists the recorded standby energy values for several time intervals.

The standby energy for 30 minutes is 106 Joules, and the consumption is relatively linear with respect to time. For 60 minutes of standby time, the energy consumed according to our measurements should be around 212 J. This result agrees with those observed in [14] whereby researchers stated that the energy consumption of the Samsung Nexus S Android phone for one hour is 216 J when in standby.

Table 5: Possible values of N

Time (min)	Standby Energy (J)
5	22
10	37
15	56
20	71
25	89
30	106

To calculate the amount of standby time gained when using SecVoIP instead of SRTP, the following formula is applied:

$$\text{Gain in Standby Time} = \frac{(TalkTime) \times (Saved\ J/\min)}{(\text{Standby}\ Energy\ J/\min) = (3.5\ J/\min)}$$

According to [32], the typical AT&T customer averaged 21 minutes per day in the first quarter of the year 2011. For a talking time of only 21 minutes per day, the gain in extended standby and talk times when using SecVoIP as compared to SRTP are as shown in Table 6.

To generalize, the formulas in Table 7 can be used to calculate the extension of standby time and talk time for any value of the overall talk time per day.

Table 6: Saving values for the 21 minute/day case

Case	Savings (J/min)	Extension in standby time (min) per day	Extension in talk time (min) per day
GSM SecVoIP N=5	1.8	10.8	8.0
GSM SecVoIP N=15	2.2	13.2	10.7
G.722 SecVoIP N=5	4.8	28.8	8.2
G.722 SecVoIP N=15	6.8	40.8	13.9

4 Conclusions

The paper presented SecVoIP, which is a proposed algorithm to overcome several weaknesses that existed in previous algorithms to secure VoIP such as E^2VoIP^2. Eliminating the need to pad additional encryption-related information to the first packet of each group was the key

solution. SRTP was selected as the benchmark protocol due to its popularity and efficiency. The algorithm is secure as long as the attacker is incapable of deducing the original plaintext data. Several experiments were conducted in this context and invariably show that the stochastic nature of the codec output prevents the attacker from recovering the original plaintext data even when eavesdropping on the victim's conversation. The efficiency of the proposed mechanism over SRTP and over E^2VoIP^2 is demonstrated using three different experiments. The degree of improvement is dependent on the voice packet size and on the number of packets per group. Finally, experimental results showed that SecVoIP outperforms SRTP in terms of battery usage and talk time.

Table 7: General extension values

Case	Savings (J/min)	Extension in standby time (min)	Extension in talk time (min) per day
GSM SecVoIP N=5	1.8	0.514 x Talking Time	0.383 x Talking Time
GSM SecVoIP N=15	2.2	0.629 x Talking Time	0.512 x Talking Time
G.722 SecVoIP N=5	4.8	1.371 x Talking Time	0.390 x Talking Time
G.722 SecVoIP N=15	6.8	1.943 x Talking Time	0.660 x Talking Time

Acknowledgments

The authors would like to acknowledge the support of the Lebanese National Council for Scientific Research and the American University of Beirut University Research Board.

References

[1] A. L. Alexander, A. L. Wijesinha, and R. Karne, "An evaluation of secure real-time transport protocol (SRTP) performance for VoIP," in *Third International Conference on Network and System Security*, pp. 95-101, 2009.

[2] R. Barbieri, D. Bruschi, and E. Rosti, "Voice over IPsec: Analysis and solutions," in *Proceedings of the 18th Annual Computer Security Applications Conference*, pp. 261, 2002.

[3] M. Baugher, D. McGrew, M. Naslund, E. Carrara and K. Norrman, *The Secure Real-time Transport Protocol (SRTP)*, RFC 3711, Mar. 2004.

[4] D. Butcher, X. Li, and J. Guo, "Security challenge and defense in VoIP infrastructures," *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, vol. 37, no. 6, pp. 1152-1162, Nov. 2007.

[5] Csipsimple Group, *SIP Application for Android Devices*, 2012. (http://code.google.com/p/csipsimple/)

[6] E. A. Charanek, H. Dermanilian, Imad H. Elhajj, A. Kayssi, and A. Chehab, "E^2VoIP^2: Energy efficient voice over IP privacy," *International Journal of Computers and Security*, vol. 30, no. 8, pp. 815-829, Nov. 2011.

[7] A. H. Cheetham and J. E. Hazel, "Binary (presence-absence) similarity coefficients," *J. Paleontol*, vol. 43, pp. 1130-1136, 1969.

[8] S. Cherry, "Winner: Sprint's broadband gamble," *IEEE* Spectrum, Jan. 2008.

[9] E. Choo, J. Lee, H. Lee, and G. Nam, "SRMT: A lightweight encryption scheme for secure real-time multimedia transmission," *Multimedia and Ubiquitous Engineering*, pp. 60–65, 2007.

[10] S. E. Diaa, A. M. Hatem, and H. M. Mohty, "Evaluating the performance of symmetric encryption algorithms", *International Journal of Network Security*, vol. 10, no. 3, pp. 213-219, May 2010.

[11] W. B. Diab, S. Tohme, and C. Bassil, "VPN analysis and new perspective for securing voice over VPN networks," in *Fourth International Conference on Networking and Services*, pp. 73-78, 2008.

[12] A. Elbayoumy and S. Shepherd, "Stream or block ciphers for securing VoIP*?" International Journal of Network Security*, vol. 5, no. 2, pp. 128-133, 2007.

[13] J. François, R. State, T. Engel, and O. Festor, "Digital forensics in VoIP networks," *IEEE International Workshop on Information Forensics and Security*, pp. 1-6, 2010.

[14] Google Docs, *Nexus S Battery Drain Benchmark*, 2012. (https://docs.google.com/spreadsheet/ccc?key=0AntDDKv-lS6IdEY4T3dXWVFDWk9lREdJNGFfU2NwRlE#gid=0)

[15] M. Gordon, *A Power Monitor for Android-Based MobilePlatforms*, 2012. (http://ziyang.eecs.umich.edu/projects/powertutor/)

[16] M. Grangetto, E. Magli, and G. Olmo, "Multimedia selective encryption by means of randomized arithmetic coding," *IEEE Transactions on Multimedia*, vol. 8, no. 5, pp. 905-917, Oct. 2006.

[17] P. Gupta, V. Shmatikov, I. VMWare, and P. Alto, "Security analysis of voice-over-IP protocols," in *20th IEEE Computer Security Foundations Symposium*, pp. 49-63, 2007.

[18] C. T. Hager, S. F. Midkiff, J. M. Park, and T. L. Martin, "Performance and energy efficiency of block ciphers in personal digital assistants," in *Proceedings of the 3rd IEEE International Conference on Pervasive Computing and Communications*, pp. 127-136, 2005.

[19] J. K. Han, H. Y. Chang, S. Cho, and M. Park, "EMCEM: An efficient multimedia content encryption scheme for mobile handheld devices" in *International Conference on Information Science and Security*, pp. 108-114, Jan. 2008.

[20] K. Hong, S. Jung, L. L. Iacono, and C. Ruland, "Impacts of security protocols on real-time multimedia communications," *Information Security Applications*, pp. 1-13, 2005.

[21] S. Karapantazis and F. N. Pavlidou, "VoIP: A comprehensive survey on a promising technology," *Computer Networks*, vol. 53, pp. 2050-2090, 2009.

[22] D. R. Kuhn, T. J. Walsh, and S. Fries, "Security considerations for voice over IP systems," *NIST Special Publication*, pp. 800-858, 2005.

[23] National Institute of Standards and Technology, *Random Number Generation*, Computer Security Resource Center, 2011. (http://csrc.nist.gov/groups/ST/toolkit/rng/index.html)

[24] P. Prahithsangaree and P. Krishnamurthy, "Analysis of energy consumption of RC4 and AES algorithms in wireless LANs," in *IEEE Global Telecommunications Conference (GLOBECOM)*, vol. 3, pp. 1445-1449, 2003.

[25] B. Prijono, *Open Source SIP Stack and Media Stack for Presence*, 2011. (http://www.pjsip.org)

[26] PTLsim, *x86-64 Cycle Accurate Processor Simulation Design Infrastructure*, 2012. (http://www.ptlsim.org)

[27] D. Qiao, M. Gursoy, and S. Velipasalar, "Secure wireless communication and optimal power control under statistical queuing constraints," *Information Forensics and Security, IEEE Transactions on*, no. 99, pp. 628-639, 2011.

[28] N. Ruangchaijatupon and P. Krishnamurthy, "Encryption and power consumption in wireless LANs," in *Third IEEE Workshop on Wireless LANs, Newton*, pp. 27-28, Massachusetts, Sep. 27-28, 2001.

[29] A. Servetti and J. C. De Martin, "Perception based selective encryption of G.729 speech," in *IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 1, pp. I-621-I-624, 2002.

[30] A. Servetti, C. Testa, and J. C. De Martin, "Frequency selective partial encryption of compressed audio," in *Proceedings of CASSP*, pp. 668-671, 2003.

[31] Thermos and A. Takanen, *Securing VoIP Networks: Threats, Vulnerabilities, and Countermeasures*, Addison-Wesley Professional, USA, Aug. 2007.

[32] S. Woolley, *Cell phone use is way up. So why did brain cancer rates fall?*, CNN, 2012. (http://tech.fortune.cnn.com/2011/06/07/cell-phone-use-is-way-up-so-why-are-brain-cancer-rates-down/)

[33] D. Xie and C. C. J. Kuo, "Multimedia data encryption via random rotation in partitioned bit streams," *IEEE International Symposium on Circuits and Systems*, vol. 6, pp. 5533-5536, May. 2005.

[34] M. Wu and Y. Mao, "Communication friendly encryption of multimedia," in *IEEE Workshop on Multimedia Signal Processing*, pp. 292-295, Dec. 2002

.

Hoseb Dermanilian received the BE degree with high distinction in Electric and Computer Engineering from Aleppo University in 2009. He holds a Masters in Engineering degree from the American University of Beirut, his research interests are in the field of computer and communications networks with emphasis on communications security, VoIP security, energy efficient security, and has to date published one paper in energy aware computing. He is currently working as an ICT Project Engineer in an Airport Expansion Project leading Computing and Storage System.

Farah Saab received the BE degree in Electrical and Computer Engineering in 2011 from the American University of Beirut (AUB). She is currently a graduate research assistant at AUB and part of the Middle East Energy Efficiency Research project. Her main research interests include energy efficient computing as well as networks and security.

Imad H. Elhajj received his Bachelor of Engineering in Computer and Communications Engineering, with distinction, from the American University of Beirut in 1997 and the M.S. and Ph.D. degrees in Electrical Engineering from Michigan State University in 1999 and 2002, respectively. He is currently an Associate Professor with the Department of Electrical and Computer Engineering at the American University of Beirut. Dr. Elhajj is the vice-chair of IEEE Lebanon Section, senior member of IEEE and senior member of ACM. His research interests include instrumentation and robotics, cyber security, sensor and computer networks, and multimedia networking with more than 100 publications. Imad received the Most Outstanding Graduate Student Award from the Department of Electrical and Computer Engineering at Michigan State University in April 2001, the Best Paper award at the IEEE Electro Information Technology Conference in June 2003, and the Best Paper Award at the International Conference on Information Society in the 21st Century in November 2000. Dr. Elhajj is recipient of the Teaching Excellence Award at the American University of Beirut, June 2011.

Ayman Kayssi was born in Lebanon. He studied electrical engineering and received the BE degree, with distinction, in 1987 from the American University of Beirut (AUB), and the MSE and PhD degrees from the University of Michigan, Ann Arbor, in 1989 and 1993, respectively. He received the Academic Excellence Award of the AUB Alumni Association in 1987. In 1993, he joined the Department of Electrical and Computer Engineering (ECE) at AUB, where he is currently a full professor. In 1999-2000, he took a leave of absence and joined Transmog Inc. as chief technology officer. From 2004 to 2007, he served as chairman of the ECE Department at AUB. He teaches courses in electronics and in networking, and has received AUB's Teaching Excellence Award in 2003. His research interests are in information security and networks, and in integrated circuit design and test. He has published more than 165 articles in the areas of VLSI, networking, security, and engineering education. He is a senior member of IEEE, and a member of ACM, ISOC, and the Beirut OEA.

Ali Chehab received his Bachelor degree in EE from AUB in 1987, the Master's degree in EE from Syracuse University in 1989, and the PhD degree in ECE from the University of North Carolina at Charlotte, in 2002. From 1989 to 1998, he was a lecturer in the ECE Department at AUB. He rejoined the ECE Department at AUB as an Assistant Professor in 2002 and became an Associate Professor in 2008. He received the AUB Teaching Excellence Award in 2007. He teaches courses in Programming, Electronics, Digital Systems Design, Computer Organization, Cryptography, and Digital Systems Testing. His research interests include: Wireless Communications Security, Cloud Computing Security, Multimedia Security, Trust in Distributed Computing, Low Energy VLSI Design, and VLSI Testing. He has about 130 publications. He is a senior member of IEEE and a member of ACM.