ISSN: 0067-2904
GIF: 0.851

A Secured End to End Voice Transmission in Smartphone

Sarab M. Hameed*, Israa Nafea Mahmood
Department of Computer Science, College of Science, University of Baghdad, Baghdad, Iraq

Abstract

The traditional voice call over *Global System for Mobile Communications* (GSM) and *Public Switched Telephone Network* (PSTN) is expensive and does not provide a secure end to end voice transmission. However, the growth in telecommunication industry has offered a new way to transmit voice over public network such as internet in a cheap and more flexible way. Due to insecure nature of the public network, the voice call becomes vulnerable to attacks such as eavesdropping and call hijacking. Accordingly, protection of voice call from illegal listening becomes increasingly important.

To this end, an android-based application (named *E2ESeVoice* application) to transmit voice in a secure way between the end users is proposed based on modified RC4. The performance of the proposed *E2ESeVoice* application is evaluated in terms of Mean *Opinion Score (MOS)* and *computation time*. The MOS score of E2ESeVoice application is measured with encrypted and unencrypted calls and resulted in values of 4.25 (Good) and 4 (Good) respectively. The result shows that the MOS of both tests is comparable and the encryption has not affect the quality of voice.

Keywords*:* RC4, Voice encryption, VOIP

نقل صوت آمن من طرف الى طرف في الهاتف الذكي

سراب مجيد حميد*، اسراء نافع محمود
قسم علوم الحاسبات، كلية العلوم ، جامعة بغداد، بغداد، العراق

الخلاصة

ان المكالمات الصوتية التقليدية عبر النظام العالمي للاتصالات المتنقلة (GSM) و شبكة الهاتف العامة للاتصالات (PSTN) مكلفة ولاتوفر خطوط امنة لنقل الصوت. ومع ذلك فان النمو الحاصل في صناعة الاتصالات قد و فر وسيلة جديدة لنقل الصوت عبر الشبكة العامة مثل الانترنت و هي وسيلة رخيصة و اكثر مرونة من الشبكات التقليدية. لكن نظرا لطبيعة الشبكة العامة التي تكون عرضة للهجمات مثل التنصت و خطف المكالمة فان حماية هذه الشبكة اصبحت ذات اهمية متزايدة في الاونة الاخيرة . تحقيقا لهذه الغاية فقد صمم تطبيق اندرويد والذي يطلق عليه (*E2ESecVoice*) لنقل الصوت بطريقة امنة وغير مكلفة بين المستخدمين ان التطبيق تبنى خوارزمية التشفير اعتمادا على خوازمية modified RC4 . تم تقييم اداء التطبيق من حيث متوسط نقاط الراي (MOS) و وقت الحساب. أن اختبار MOS الذي تم تجربته للمكالمة المشفرة و الغيرة مشفرة توضح بان النتائج متقاربة و ان تشفير الصوت لم يؤثر على نوعية الصوت.

*Email: sarab_majeed@yahoo.com

1. Introduction

The development in telecommunication industry especially in wireless network and wireless devices from simple phone to smartphone has led to the use of wireless devices for important communication such as *electronic mails* (emails), business communication activities, video conference, data and voice communication.

Moreover, smartphone devices such as Android and iOS smartphones have replaced the role of traditional computers and become the most popular devices in both personal and business lives. The smartphone devices can use both *Wireless Fidelity* (Wi-Fi) and cellular networks. In general, voice traffic is carried on cellular network and recently the Wi-Fi network is used to carry voice in addition to the data[1].

The transmission of voice, which has been studied for over 100 years, is still a high-investigated topic nowadays. In the early days of telephony, transmitting voice via dedicated communication medium called *circuits* provides guaranteed services. In parallel with the existence of the *Circuit-Switched* (CS) telephony systems, a new technology was developed to transmit voice using packet-switched network such as *Voice over Internet protocol* (VOIP) [2].

The systems that use packet-switched network to transmit voice work by converting the human voice that is defined as an analog signal into digital signal. Then, it is packetized into *Internet Protocol* (IP) and sent over the network to the other end of the call. Transmission of voice using packet switched network is threatened by the fact that if there is a third person listening to the voice conversation without the knowledge of the caller and recipient.

The security issue of the wireless network is considered as the major drawback. Since the data while being transmitted becomes vulnerable to different types of attacks such as *eavesdropping, denial of service,* and *man in the middle attack.*

Several techniques are raised to eliminate these attacks. At the heart of them is *encryption*, which can be defined as the technique of altering information so to anyone other than the intended receiver it will look like meaningless data. When the information reached the receiver, it needs to be decrypted that is, turned back into the original message by the receiver, and only by the receiver[2].

In this paper, an Android application that provides end to end secure voice transmission coined as (E2ESecVoice) is developed to eliminate the threat of voice traffic interception within *Local Area Network* (LAN) without affecting the quality of the voice.

The rest of the paper is organized as follows: In Section 2, a related work is presented. In section 3, we introduce the proposed E2ESecVoice application. Section 4 analyzes the results. Finally, section 5 concludes the work of this paper.

2. Related Work

Several techniques have been developed over the years in order to create a secure multimedia transmission system that transmit images, audio and video in a secure form over different networks, some of these techniques are as follow :

Qi *et. al.* in 2008 [3] developed a new way to secure voice in *Global System for Mobile Communications* (GSM) networks by adding more encryption. The new approach solves the issue that traditional encryption algorithms cannot be used in voice channel directly because of *Regular Pulse Excitation-Long Term Prediction* (RPE-LTP) vocoder requirements in GSM System. The new approach achieves secured end-to-end communications in the GSM system with short time delay and good compatibility to all GSM networks.

Massandy and Munir in 2012 [4] presented an android application that transmits video in a secure way. The application captures video from smartphone camera and then transmits it to the computer in real time. The video can be played using *Video LAN Client* (VLC) video player. Selective encryption was used to encrypt only important parts of video with *Advanced Encryption Standard* (AES) Encryption algorithm.

Chumchu *et. al.* in 2012 [5] proposed a framework for end to end voice encryption over GSM network. The prototype supports real time and full duplex communication. The prototype is implemented with Bluetooth communication that is used to communicate with the mobile phone. The proposed framework used *Rivest Cipher 4* (RC4) encryption algorithm to encrypt the voice. The result shows that the prototype with Bluetooth communication is significantly secured and has an acceptable performance.

Al-hazaimeh in 2012[6] proposed a new cryptographic encryption to secure voice over internet protocol that minimizes the delay of packet transmission over internet (i.e. packet latency and loss due to the nature of IP Networks). The proposed method used a simple and strong encryption algorithm as well as an embedded method to exchange the keys between the users. Linear Feedback Shift Register (LFSR) is used to generate pseudo random keys with a variable bits size and RC6 is used to achieve confusion and diffusion operations. The results realizes that the proposed method can be used in real time application VoIP calls.

Ahmed *et. al*. in 2012 [7] proposed encryption technique based on Lorenz and Chen chaos systems [8] to protect the voice from illegal listening. The mixed chaos generates encryption key that should be unpredictable and random. The generated key is masked with voice bit-stream to produce ciphertext. The proposed technique proves its effectiveness in terms of signal distribution and auto-correlation.
Dodmane and Aithal in 2013[9] presented a two-level security system to encrypt audio that involves both transposition and substitution cipher. First, the audio is encrypted with transposition cipher. Then, Modulus Multiplication is used. The key is generated using *Pseudo Random Number Generation* (PRNG). The proposed system provides secure transmission for audio with short amount of delay, which indicates it applicability to be used for real time application.

3. The proposed E2ESecVoice application

An android based end to end secure voice transmission application coined as (E2ESecVoice) is developed to protect call confidentiality. E2ESecVoice application supports full duplex communication and exploits Wi-Fi technology to communicate between the two clients in a LAN Network. Furthermore, modified RC4 algorithm is adopted to preserve the privacy of digital of voice [10]. Moreover, E2ESecVoice application simulates the work of VOIP to transmit voice using *User Datagram Protocol* (UDP) instead of the VOIP protocols. The proposed E2ESecVoice application can be installed on 2.x android version and higher. At the sender side, the E2ESecVoice sends a signal to the Microphone to start recording. The Analog to Digital (A/D) converts the voice from analog to digital and stores it in memory using temporary buffer. Then, the content of the buffer is encrypted and transmitted through Wi-Fi to the receiver. At the receiver side, the data that is delivered through Wi-Fi and stored in buffer and the E2ESecVoice decrypts the data and send it to the speaker after converting it from Digital to Analog (D/A).

In addition, the proposed system contains a server to manage the connection among the clients and distributes the secret keys. The server is always up and running to handle requests from client. When a client runs the application, the client application sends an acknowledgement to the server. The server is connected to a database that stores Client ID, the dynamic Internet Protocol (IP) and the public key for each client. When a client connects to the network and opens up the application, the dynamic IP for the client in the database is updated to a new IP. The calling scenario protocol, as depicted in Figure-1, allows establishing the secure connection between the clients by negotiating the secret keys to be used to protect the transmitted voice.

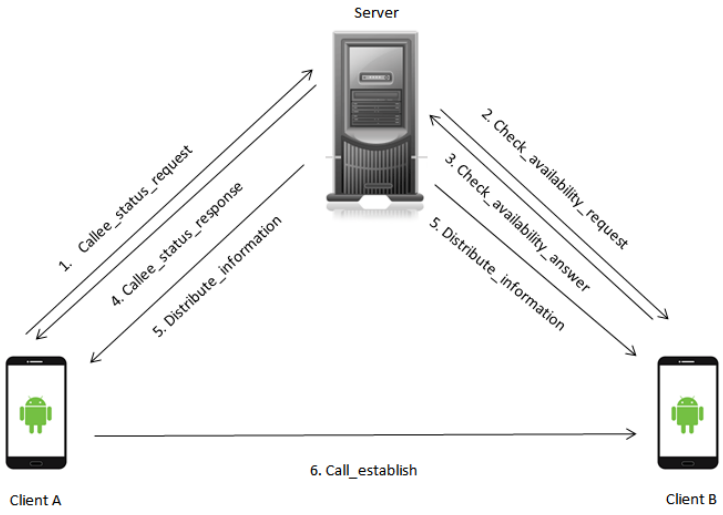


Figure 1-The Calling Scenario Protocol

The calling scenario initiated by the caller (e.g. client A) that sends callee_status_request with callee ID parameter to the server to call client B. The server checks the availability of the callee (i.e. client B) by sending check_availability_request message to client B and waits for an answer. If client B is online, then the server sends IP of client B to client A and sends the secret keys that are encrypted by elliptical curve cryptography [11] to client A and client B. Finally, a secure connection between client A and client B is established. Otherwise (i.e. client B is not available) an error message is sent to client A. Table-1 demonstrates the message type exchanged by the server and E2ESecVoice application.

Table 1- Message Type and Parameter

Message Type	Parameters
callee_status_request	Callee ID
check_availability_request	Null
check_availability_Answer	Null
callee_status_response	Callee status
Distribute_information	Encrypted secret keys, IP of the client
Call_establish	Null

E2ESecVoice application addresses the interception of voice traffic over Wi-Fi by encrypting voice calls. E2ESecVoice application consists of three stages. Figure-2 depicts the stages of the *E2ESecVoice* application.

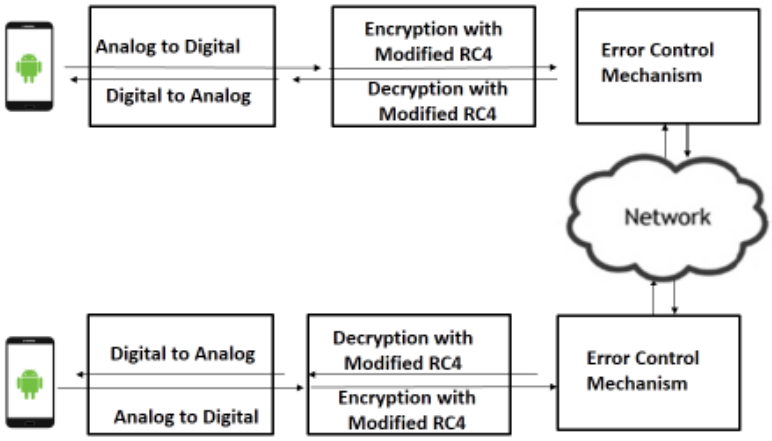


Figure 2-E2ESecVoice Application Stages

Firstly, in E2ESecVoice application, the voice is converted from analog to digital by digitizing the voice and packetizing it in small chunks. Then, these chunks are encoded and encrypted using modified RC4 The modified RC4 algorithm is adopted because, as mention in [10], it overcomes the weakness of the key scheduling algorithm of the original RC4, enhances the secrecy of the ciphertext and the time of encryption of both algorithms is comparable. After that, an error control mechanism is applied. Finally, the packets are transmitted in UDP packets through wireless network. At the destination, the E2ESecVoice application rearranges the packets. Then, the packets are decrypted and decoded. Finally, the packets are sent to the speaker to reconstruct the speech.

3.1 Analog and Digital Streams

Handling voice by converting it from analog signal to digital and vice versa is considered as the major stage in the development of E2ESecVoice application.

The voice that comes from the Mic is recorded. There are two API classes in Android Software Development Kit (SDK) to record audio namely AudioRecord class and MediaRecorder class.

The MediaRecorder is easy to work with, but the possibilities to modify the data are limited e.g., the output of MediaRecorder is an audio file in a specific file format (raw data file are not possible). In contrast, AudioRecord is more sophisticated but the user can perform mathematical operations such as encryption. The AudioRecord class provides a sampled version of the recorded audio that is saved in temporary variable and can be accessed and manipulated during the application running time. In addition, the AudioRecord controls the resources for java application to record Audio from Microphone.

One of the main characteristics of the proposed E2ESecVoice application is to be constrained with time (i.e. Operated in an online mode). To this end, AudioRecord class is adopted in this study instead of MediaRecorder. Several parameters should be initialized before using the AudioRecord class:

1. Audio Source (AS) chooses the audio source that is used to record voice in AudioRecord. In this Application, the audio source is set to MIC
2. Sample Rate (SR) represents the total number of samples that are taking from the audio per second.
3. Channel Configuration ($Channel_{Config}$) represents the audio channel configuration, which can be mono or stereo. In this thesis, the mono configuration is used because according to android is guaranteed to work on all devices.
4. Audio Format ($Audio_{Format}$) selects the digital format coding for the analog audio data. In android, there is an API for encoding audio data that is recorded and stored in the device such as *Pulse Code Modulation* (PCM) in which each sample can be represented by 8 or 16 bits.
5. Buffer Size ($Buffer_{Size}$) is considered the most important factor to successfully initialize the AudioRecord class.

It is important to point out that the incorrect set for buffer size can cause an error such as crash of the application and noise generation. This motivates us to investigate how to properly set the size of the buffer by examining the relation between sample rate and buffer size. After trying out different values for the buffer size, the relation to set an enough buffer size for a given sample rate is suggested to be $Buffer_{Size} < \frac{Samaple_{Rate}}{2}$. The lower bound of buffer size can be set to any value. However, we have to consider that the buffer size must be acceptable with our system design (i.e. the resources of the system). For example, the buffer size can be equal 32, 64 or 128 but decreasing buffer size leads to increase the used resources and cause an overload to the network.

At the receiver side, the data is converted from digital to analog. The AudioTrack class is used to play out the voice through speaker. The configuration of AudioTrack should be the same as the configuration of AudioRecord with the addition of the mode of the AudioTrack.

The AudioTrack mode can be static or stream. The static mode is used when the audio source is short and can be accommodated in memory while the stream mode is used when there is a continuous stream of data. In E2ESecVoice application, AudioTrack in stream mode is used since the phone conversation can be defined as continuous stream of audio.

3.2Error control Mechanism

Since the time factor plays an important role in E2ESecVoice application therefor, UDP in transport layer is used to transmit the voice. Voice is digitized and packetized in small chunks of 20 or 30 milliseconds (ms). Then, it is transmitted to the receiver through packet switched network. Therefore, in case of lost packet, the application can handle the loss of 30 ms of the conversation instead of waiting for a few seconds in order to retransmit the lost packet. Moreover, the network should provide high bandwidth that assures the minimum amount of packet loss.

E2ESecVoice transmits secure voice using symmetric key algorithm namely modified RC4 [10] for voice encryption. Therefore, the application must guarantee that the key used to encrypt the voice data is the same as the key used to decrypt the packet. Otherwise, the use of an incorrect key to encrypt the packet causes serious situation where the voice packets will not be decrypted at all and the E2ESecVoice application will not able to reconstruct the speech again. Accordingly, an error control mechanism is suggested to ensure that the sent and received data are encrypted and decrypted with the same key.

In error control mechanism, one-byte as a sequence number counter (S_{Count}) is appended to the encrypted voice data at sender side before sending it to the receiver through LAN. The sequence number counter provides the receiver the ability to detect if there is a lost or delayed packet. S_{Count} is initialized to zero by the sender. Each time that a data is sent, the S_{Count} is incremented. Sequence number counter may not exceed $2^8 - 1$.

At the receiver side, the received encrypted voice data should be decrypted and decoded to reconstruct the speech. The receiver can ignore the lost packet and drops the delayed packet depending on sequence number counter.

The basic idea of the suggested error control mechanism is that the sender can send the data at whenever he/she wants, the receiver does not send an acknowledgment, a receiver is ready to receive any packet and the sender does not retransmit lost. Moreover, the receiver counter (R_{Count}) is initialized to zero and incremented by one when a data is received. Furthermore, once the counter reach 255, the counter is reset to zero. In what follow, three situations are considered to clarify the error control mechanism.

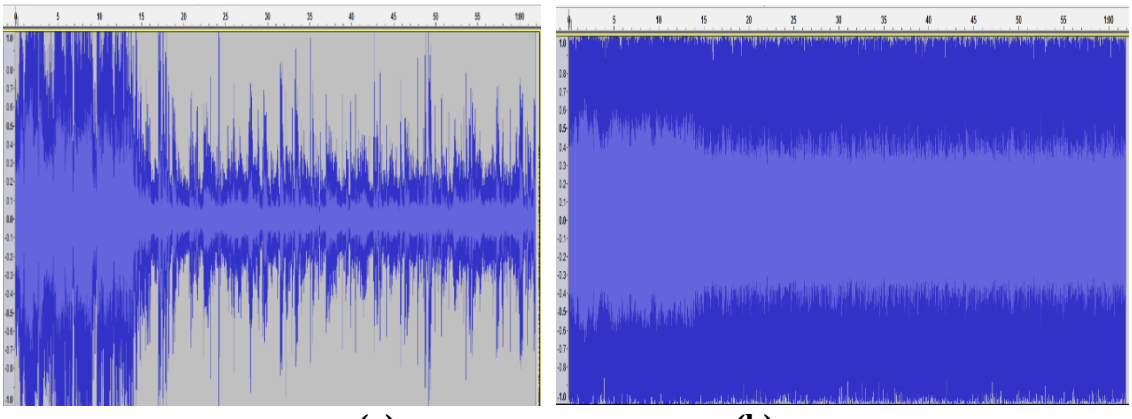
1. Normal Situation: the data is successfully transmitted with no problem. The receiver checks the sequence of the received data S_{Count}. If the sequence number equals to R_{Count} then this means that a successful delivery of data is conducted and R_{Count} is incremented by one.
2. Packet Loss Situation: the receiver checks the sequence number of the delivered data, S_{Count}. If the sequence number is greater than the R_{Count} (i.e. a packet is lost) then the receiver should resynchronize the key with the sender and R_{Count} is incremented by 2.
3. Packet Delay Situation: for example, the receiver receives Voice Payload2 and Voice Payload1 is delayed. The receiver checks the sequence number of delivered data. If the sequence number is smaller than R_{Count} (i.e. a packet is delayed) then the receiver simply drops the packet.

4. Performance Evaluation

Two criteria can be used to measure the voice quality of E2ESecVoice application namely Mean Opinion Score (MOS) and computation time. MOS is considered as one of the widely used subjective voice quality measurement method. A number of listeners evaluates the voice quality for a voice call. The MOS score ranged from 1 to 5, where 1 means “bad” , 2 means “poor", 3 means “fair", 4 means “good" and 5 means "excellent" [12].

4.1Voice Encryption with Modified RC4

Figure-3 depicts the waveform of unencrypted and encrypted calls with modified RC4 for 1-minute call duration respectively. As shown in the figures there is no correlation between encrypted and unencrypted calls that means the modified RC4 makes the call unpredictable and it is difficult to realize the call.



(a) **(b)**

Figure 3-Waveform of (a) Unencrypted Call (b)Encrypted Call with Modified RC4

4.2Impact of Buffer Size

Multiple factors should be taken into account during the process of choosing the buffer size such as processor speed and sampling rate. The buffer size is essential in determining the amount of delay.The impact of buffer size is analyzed under five different settings, 512 $bytes$, 1 $kilobyte$ (kB), 2 kB, 4 kB and 8 kB and four different sampling rate in kilohertz (KHz) 8, 16, 22.05 and 32. Figure-4 depicts the

impact of the buffer size and sampling rate on mouth to ear round trip time $(M2ERTT)$ in (ms). $M2ERTT$ is defined as the amount of time between the caller speaks a word and the callee hears the word and sends an acknowledgment for that word to the caller. The result reveals that reducing buffer size and increasing sampling provides a less amount of round trip time but at the same time additional computational power is required. The use of smartphone, which has limited resources, affects the choice of buffer size. The number of packets transmitted when sampling rate equals 32 KHz for one minute has increased five times comparable to the number of packets transmitted when sampling rate equal 8 KHz. This causes an overhead on both the network and the device computational power.

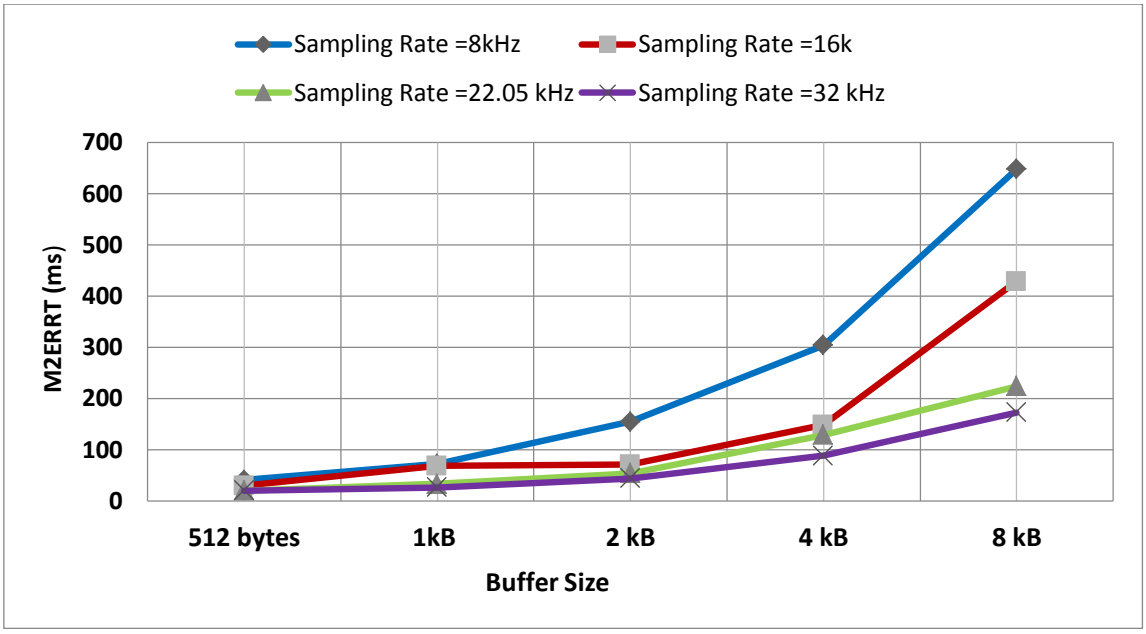


Figure 4- $M2ERTT$ with Different Settings of Buffer Size and Sampling Rate

Figure-5 depicts the CPU usage of E2ESecVoice application while varying number of sampling rate and for five different settings of buffer size. $M2ERTT$ can be decreased by increasing sampling rate and decreasing buffer size but at the expense of increasing CPU usage. Therefore, the E2ESecVoice application uses 8kHz sampling rate and 512 bytes buffer size to balance between $M2ERTT$ and the CPU usage.

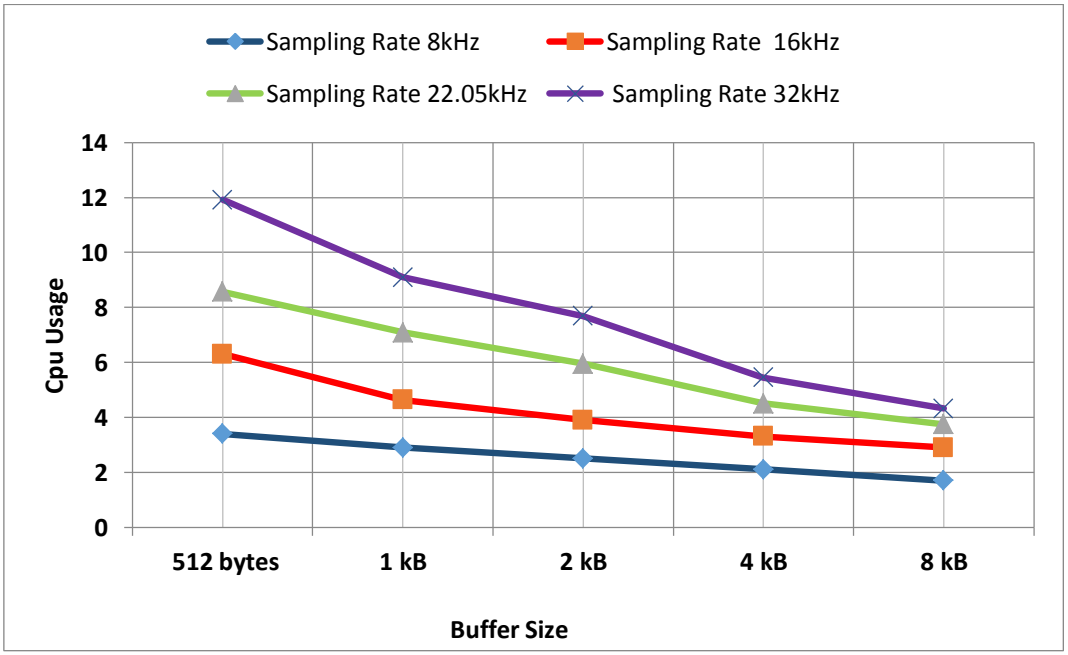


Figure 5-E2ESecVoice CPU Usage with Different Settings of Buffer Size and Sampling Rate

4.3MOS of E2ESecVoice Application

The quality of voice is measured subjectively in terms of MOS. The E2ESecVoice MOS is tested over 30 listeners. The listeners are chosen with ages from 20-60 years old and 15 female and 15 male. Furthermore, the 30 listeners are divided into three groups each group contains 10 listeners. The first group listens to 3 minutes of unencrypted call and encrypted call. The second group listens to 6 minutes of unencrypted call and encrypted call. Finally, the third group listens to 9 minutes of unencrypted call and encrypted call. The MOS value is the mean of the marks of all listeners in each group. The MOS of unencrypted call and encrypted call is illustrated in Table-2. The results clarify that the difference between unencrypted call and encrypted is very small which means that the adopted modified RC4 has no obvious effect on voice quality.

Table 2- The MOS of Unencrypted and Encrypted Calls

Group#	MOS	
	E2ESecVoice without Encryption	E2ESecVoice with Encryption
1	4.25	4
2	4.1	3.9
3	4.15	3.9

Furthermore, as stated in the previous section that the size of the buffer affects $M2ERTT$. Therefore, the MOS is measured when the listeners listen to encrypted call duration equals 1 min for different buffer size with varying sampling rate. Figure-6 depicts MOS of E2ESecVoice application when sampling rate =8 kHz and buffer size equals 512 $byte$, 1 kB, 2 kB, 4 kB, or 8 kB. Figure-7, -8 depict MOS of E2ESecVoice application when sampling rate equals 16 kHz and 22.05 kHz with buffer size equals 1 kB, 2 kB, 4 kB, 8 kB, or 16 kB respectively. Figure-9 shows the MOS of E2ESecVoice application when sampling rate equals 32 kHz and buffer size equals 2 kB, 4 kB, 8 kB, 16 kB or 32 kB. As can be seen in the Figures when buffer size is less than $sampling\ rate/2$, MOS is satisfied with an acceptable degree. On the other hand, when buffer size is equal to or greater than $sampling\ rate/2$, a decreasing in MOS can be observed. This comes from that increasing buffer size, the M2ERTT is also increased (i.e. the time required to receive the voice from the sender is increased) and that delay affects the quality of voice.

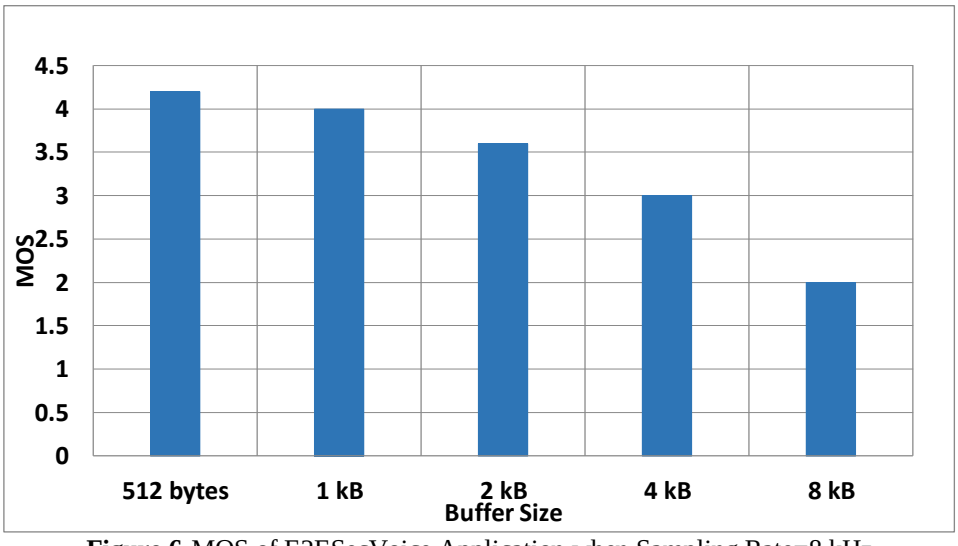


Figure 6-MOS of E2ESecVoice Application when Sampling Rate=8 kHz

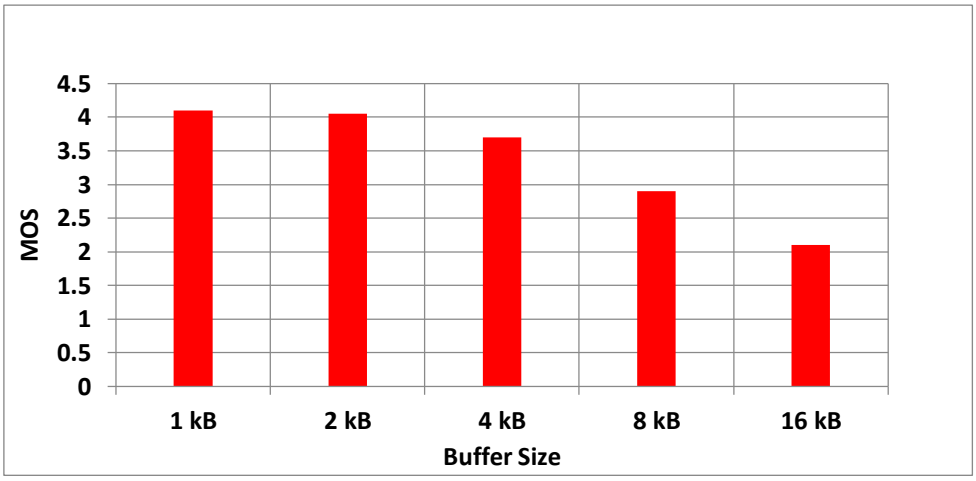


Figure 7- MOS of E2ESecVoice Application when Sampling Rate=16 kHz

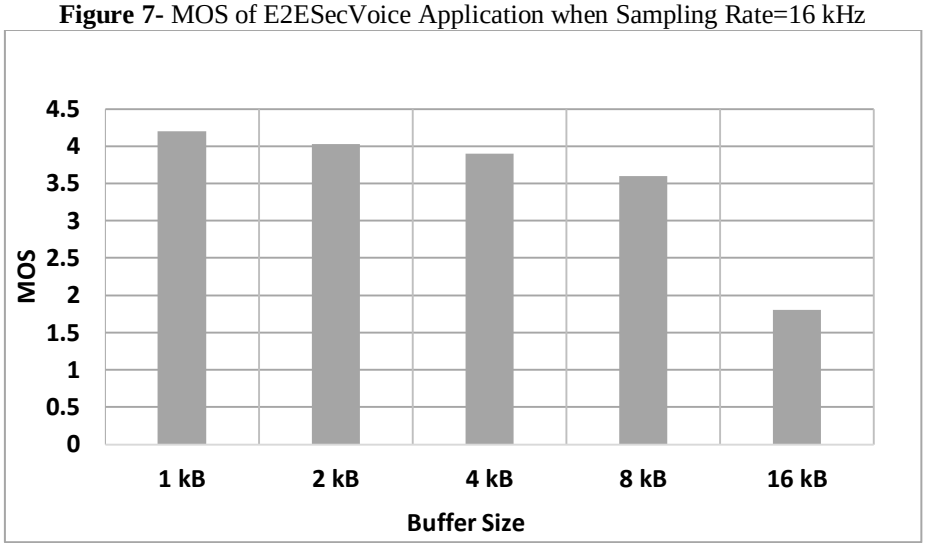


Figure 8- MOS of E2ESecVoice Application when Sampling Rate=22.05 kHz

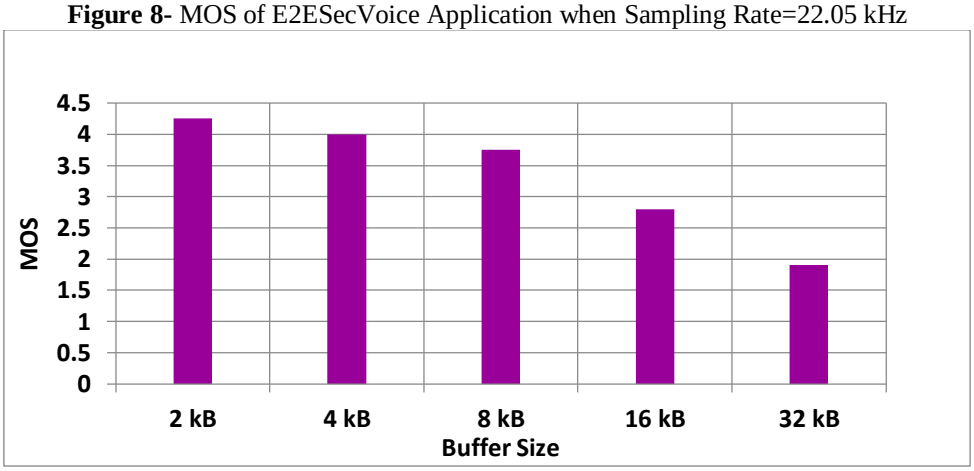


Figure 9- MOS of E2ESecVoice Application when Sampling Rate=32 kHz

4.4 Computation Time of E2ESecVoice Application

The voice quality should be measured objectively in terms of mouth to ear latency that is defined as the amount of time between the caller speaks a word and the callee actually hears that word [13]. However, $M2ERTT$ is used to measure the voice quality due to the difficulty of synchronizing the time between the caller and callee. Table-3 shows the mouth to ear round trip time for the unencrypted and encrypted call for different call durations.

From the table, one can see that the average $M2ERTT$ in 3 min, 6 min and 9 min call durations is 41.3ms , 41.69 ms and 41.16 ms respectively if no encryption is used. While the average $M2ERTT$ is 44.7ms, 43 ms , and 45.52 ms in 3min, 6 min and 9 min call durations respectively when

encryption is used. The mouth to ear round trip time in the E2ESecVoice application is acceptable according to International Telecommunication Union-Telecommunication (ITU-T) recommendation [13] that states the delay less than 150 ms is acceptable for most application, the delay between 150 to 400 ms is acceptable and delay greater than 400 ms is unacceptable. This means the developed application is able to transmit a secure voice without affecting its quality.

Table 3- $M2ERTT$ of Unencrypted and Encrypted Calls

Call Duration (min)	$M2ERTT$	
	E2ESecVoice without Encryption	E2ESecVoice with Encryption
3	41.3	44.7
6	41.69	43
9	41.16	45.52

5.Conclusions

In this paper, a modified RC4 algorithm is adopted to fulfill secured end to end voice transmission in smartphone with an acceptable level of quality, short time delay and without requiring additional computational power. Based on the results, it can be concluded that the developed application is able to provide security for voice transmission without affecting the quality of voice. The work in this paper can be extended to include a method for synchronizing the time between the caller and called devices.

References

1. Aloraini B.S.**2014**. A New Covert Channel over Cellular Voice Channel, M.Sc. Thesis, Department of Networking, Security, and Systems Administration, Computing and Information Sciences, Rochester Institute of Technology, USA.
2. Kukkar S. **2012**. Secure Voip Call on Android Platform. *Global Journal of Computer Science and Technology Network, Web & Security*, 12(12).
3. Qi H.F., Yang X.H., Jiang R., Liang B. and Zhou S.J. **2008.** Novel End-to-End Voice Encryption Method in GSM System, IEEE International Conference on Networking, Sensing and Control, pp:217 - 220.
4. Massandy D. T. and Munir I. R. **2012**. Secured video streaming development on smartphones with Android platform, 7th Ineternational Conference on Telecommunication Systems, Services, and Applications., pp:339–344.
5. Chumchu P., Phayak A., and Dokpikul P. **2012**. A simple and cheap end-to-end voice encryption framework over GSM-based networks, Computing, Communications and Applications Conference, pp:210–214.
6. Al-hazaimeh O. M. A. **2012**. Increase the Security Level for Real-Time Application Using New Key Management Solution, *International Journal of Computer Science Issues*, 9(3), pp:240–24.
7. Ahmed M., Alam B. and Farooq O. **2012**. Chaotic Masking of Voice Bitstream Using Mixed Sequences for Authorized Listening, *Advances in Computer Science and Information Technology*, pp:604-611, Springer.
8. Kocarev L. **2011.** *Chaos-based Cryptography: Theory, Algorithms and Applications*, Springer.
9. Dodmane R. and Aithal G. **2013**. Efficient Audio Encryption Algorithm For Online Applications Using Transposition And Multiplicative Non-Binary System, I*nternational Journal of Engineering Research & Technology*, 2(6), pp:472–477.
10. Hameed S. M and Mahmood I.S. **2015**. Modified Key Scheduling Algorithm for RC4, *Accepted in Iraqi Journal of Science.*
11. Stallings W.**2011**. *Cryptography and Network Security principles and Practice*, Fifth Edition, Prentice Hall.
12. Ismail N. M. **2009.** Analyzing of MOS and Codec Selection for Voice over IP Technology, *Annals Computer Science Series*, 7(1), pp:263–276.
13. Agastya C., Mechanic D. and Kothari N. **2009**. Mouth-To-Ear Latency in Popular VoIP Clients, *Columbia University Computer Science Technical Reports*.