VFAST Transactions on Software Engineering
http://vfast.org/journals/index.php/VTSE@ 2019, ISSN(e): 2309-6519; ISSN(p): 2411-6327
Volume 14, Number 1, January-April, 2019 pp. 01-**05**

DESIGN AND DEVELOPMENT OF SECURE MOBILE COMMUNICATION OVER GSM NETWORK USING OPEN SOURCE OPERATING SYSTEM (OS)

ARSLAN ASIM*[1], MUHAMMAD ASHRAF[2]
[1]Institute of Aviation Studies, University of Management and Technology, Lahore, Pakistan.
[2]PAF College of Aeronautical Engineering, National University of Sciences and Technology, Risalpur, Pakistan
Email: arslan.asim@umt.edu.com

ABSTRACT. *With the rapidly advancing technology of today, exchange of information and data is a very pertinent matter. The world has just recently witnessed the effects of information leakage through the issue of WikiLeaks. There are huge amounts of data being shared over different platforms nowadays. Global System for Mobile Communication (GSM) is one of the most reliable platforms known to and used by almost all people in the world for text as well as voice communication. With the tools like Android Studio and NetBeans available, it is now possible to encrypt the text that has to be sent over the GSM, so that it can be decrypted at the other end of the communication path. However, the encryption and decryption of voice being transmitted over the GSM network still remains a question. In the domain of real time voice encryption, much of the work being carried out pertains to the voice being exchanged through the Internet Protocol. As compared to the Voice over Internet Protocol (VoIP), voice over the GSM network has not seen much research work related to its security aspects. The purpose of this paper is to document the results of a project aimed at developing a platform for mobile phones in order to communicate over the GSM network in a secure manner. The most suitable method for achieving the above mentioned objective is to use an open source Operating System (OS), so that the source code is easily accessible and usable. In this paper, the Android OS will be under discussion, which is compatible with all the Android mobile phones. In this way, the maximum number of mobile phone users can be benefitted because Android cell phones are being widely used nowadays. The use of cryptographic algorithms for securing the voice communication over the GSM network is also a part of this paper. The work revolves around the Java programming language since the Android application development has been carried out in Java through the use of Android Studio. Also, NetBeans has been employed for developing algorithms for voice encryption.*

Keywords: Android Operating System; Global System for Mobile Communication (GSM); encryption; decryption; cryptography.

1. **Introduction.** The prime objective of this paper is to analyse the voice communication over the Global System for Mobile Communication in terms of the security of the data being transmitted. Secure communication is fundamentally aimed at creating a barrier between information assets and threats. It comprises the following three aspects:

Confidentiality: Whenever sensitive information is being exchanged, it is deemed necessary to let only the authorized parties have access to it since sharing it with other people may have catastrophic results. This aspect of information security falls under the category of confidentiality [1].

Integrity: While data is transmitted from one end to the other end, it is important to make sure that no intruder is involved in the communication. In simple words, the data must not be tampered through malicious attacks or other ways. The protection of data in a way that it remains in the same form at both the sender and the receiver ends can be considered as protecting the integrity of the data [2].

Network availability: In certain cases, the user may have to face a hindrance while accesing the communications medium. This may occur because of a Denial of Service (DoS) attack through medium saturation and in other cases, through jammed communications by malicious parties. Therefore, it is important to make sure that the network is available at all times in order to ensure secure communication [3]. Therefore, any study on information security is mainly focused at these three aspects.

2. Encrypted Call Application. WhatsApp is one of the most commonly used Android applications nowadays. A striking feature that the WhatsApp offers is its end-to-end encryption. However, applications like the WhatsApp are based on Voice over Internet Protocol (VoIP). Following the example of WhatsApp, a calling application was built and different encryption techniques were tried by virtue of it in order to achieve encrypted voice communication over GSM. The Advanced Encryption Standard (AES) was used to encrypt different Android permissions. The AES algorithm used was Java based, which was why the software platforms used for this project were NetBeans and Android Studio. Initially some problems were faced in the compilation of the code, especially while implementing the AES algorithm and the android permission encryption/decryption in the same Java file. This issue was resolved by implementing the algorithm through the use of three Java files, namely AESAlgorithm.java, AppEncryption.java and AppDecryption.java. The purpose was to separate the AES algorithm from both the encryption and decryption processes as well to separate the encryption and decryption processes. This resulted in the successful compilation of the code. The application was run on different Android virtual devices. However, no prominent effect was observed on voice communication when a call was made by using an Android smart phone. Since the encryption of this Android permission did not result in real time voice encryption, it was tried to encrypt the 'Mic' permission available through the Android Studio Manifest File, but the permission was not available in the software for the given application. A possible reason for the 'Mic' permission not being accessible in the Android Studio was the version of the Android operating system being used. The idea of encrypting the Mic permission was derived from observations made regarding the calling application running on the Nougat version of the Android OS. A difference between that application and the one built in this project was of Android permissions under use.

3. Sound Wav File Encryption. After an attempt had been made to tamper real time audio, a different approach was followed.

In this approach, a sound recording application was made. A Java application as well as an Android Studio application was built. The Android Studio application was successfully compiled and run on both the Android emulator and the Android smart phone. An important feature of both of these applications is that they store the audio in wav format.

After this, the wav format file was read through a NetBeans Java application and bytes were formed from the wav format audio.

The byte arrays were made to pass through an AES open source algorithm. Two different methods were used for this approach.

A. Method 1 – Complete Wav File Encryption. Wav format audio file was created and stored by the sound recording application. It was initialized in the NetBeans Java compiler. In this approach, the complete file was read at one time and all the bytes were extracted in one go. Then, Open Source Advanced Encryption Standard (AES) algorithm was implemented in Java on the array of bytes.

Application was successfully built and run both on the Android emulator as well as Android smart phone.

Random Integers were generated as a result of encryption:

[116, 48, -55, 122, -124, 107, -49, 107, 117, -103, 126, 105, -90, 14, 35, -65, -39, 70, -49, 69, 46, 58, -56, 79, 105, -56, 56, -63, 100, 71, -92, 62, 38, 116, 24, 16, 87, -76, -127, 64, -51, 39, 11, -36, -29, -54, 59, 42, 14, 9, 109, 88, 112, -30…………]

However, the sequence of bytes was observed to be too long for the application to execute. Secondly, the bytes appear in an unfathomable form. Symbols were being used to represent bytes.

*B. Method 2 – Limited Number of Bytes from the Wav File.*In this approach, all other techniques were same except that the number of bytes to be used for encryption were controlled and limited within the program. It was tried to convert the bytes into an understandable form. Characters were generated from the bytes read/ extracted from the wav file. Attempt was also made to regenerate the audio wav format file after decryption.

Random integers were generated:

[62, -103, 77, -38, 16, -98, -105, 127, -99, 39, -121, 12,
-110, -5, -55, 72, -59, -67, -128, -34, 125, -49, 23, -57,
-59, 40, -110, -20, -102, 53, -43, ………]

As a result of this method, the number of integers generated were limited because the number of bytes to be encrypted were limited too. For example, the random integers generated from 20 bytes were as follows:

[52, 36, -9, 2, -24, 74, 79, 77, -56, -1, 69, -43, 42, 34,
81, -126, 41, 16, -30, -50, 32, 125, 14, -58, -21, -64, -
91, 91, 94, 8, 44, -95]

The integers generated by encrypting 35 bytes were as shown:

[73, -51, -44, 33, -46, -97, -95, 43, 121, -6, -119, 23,
33, 14, -46, -99, 105, 104, 73, -60, -73, 79, 78, -50, 96,
-105, - 98, -38, 92, -42, -3, -15, 19, -69, 51, 106, -1, -
128, 22, 74, 51, -85, -1, 109, 17, 107, 83, 82]

The sequence of integers generated by encrypting 100 bytes was:

[9, -2, 9, 126, -91, 61, 111, 50, 52, 19, 87, -40, -32, -90, 21, 96, -98, 79, 93, 105, -31, -56, -2, 67, 76, 66, -28, -122, 120, -110, -70, 116, 112, -10, 54, 43, -73, 53, -31, 49, 90, -84, -60, 32, 99, 106, 4, -125, -81, 94, -58, 103, -54, 115, -13, -49, -112, 46, -49, -82, -28, 103, 37, 25, 104, -71, -42, 122, -85, -8, 55, -65, -103, -57, -115, -19, -62,………………..]

This shows that each time the program is run, a unique sequence of integers was generated, with the number of integers varying because of the variable number of bytes.

4. Messaging Application. After the encryption of the audio wav file, it was necessary to develop a mechanism by which the encrypted data can be exchanged through the GSM network. Another important aspect was the decryption of the data at the other end of the communication path. To meet the former objective, a text messaging application was developed. The encrypted data consists of random integers, which can be treated as text despite the fact that those numbers have been derived from recorded audio. The application was successfully built and run on both the Android emulator for Galaxy_Nexus_API_24 and actual Android smart phone.

5. Summary. The initial aim was to develop a single platform that could capture the voice being transmitted over the Global System for Mobile Communication (GSM) and tamper it in a way that additional security was added to it. It is worth mentioning that security ciphers are already implemented on the communication being conducted over the GSM network. These include the A5/1, A5/2 and A5/3, but attempts at deciphering the algorithms pertaining to these ciphers have been successfully carried out. This raises a pertinent question that if these algorithms can be cracked, does GSM provide us with a foolproof system of communication, which is secure from all aspects? Therefore, it becomes necessary for GSM users to devise a system of communication which provides indigenous security through GSM.

As a result of this work, a successful approach has been concluded for secure voice communication over the GSM. Further work can now be carried out for developing this approach into a viable, real time voice communication solution for GSM users. The work suggests that a suitable platform for secure GSM voice communication should first record the audio message to be transmitted. For example, in this work, the audio message was recorded in the wav format. The recorded audio file can then be encrypted through an appropriate cryptographic algorithm. GSM can then be used to transmit this encrypted data.

At the receiving end, only the text i.e. the random integers generated from the actual audio file will be received after which the process of decryption can be carried out according to the specific algorithm used. Once the process of decryption is complete, the regenerated bytes can then be used to create an audio file.

At this point, a topic of utmost concern would be the quality of the voice generated. Also, using this approach would present a drawback of time delay. Whilst developing this approach, it would be necessary to keep the time delay to a minimum.

The algorithm used for this work was the Advanced Encryption Standard. After the proposed suggestions are implemented, the choice of the cryptographic algorithm would have to be sorted out. An appropriate algorithm, preferably a stream cipher, will have to be chosen for live voice encryption.

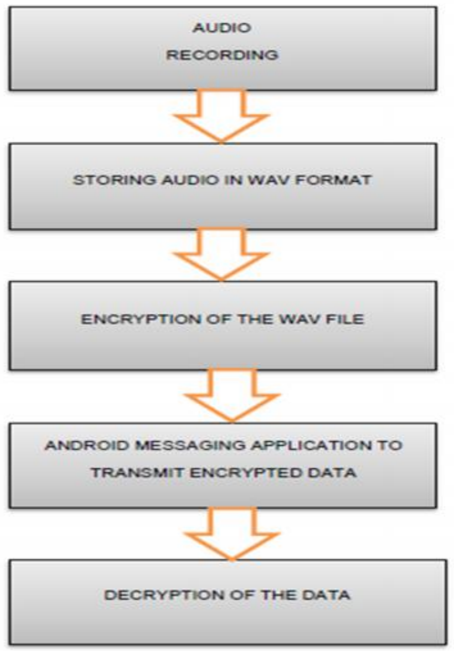
AUDIO
RECORDING
STORING AUDIO IN WAV FORMAT
ENCRYPTION OF THE WAV FILE
ANDROID MESSAGING APPLICATION TO TRANSMIT ENCRYPTED DATA
DECRYPTION OF THE DATA

Figure-1: Summary of the suggested conclusions

6. Problems Related To Real Time Audio Encryption Over The GSM Network. It is important to incorporate in this work some of the key issues that can hinder the progress towards real time encryption and transmission of voice being exchanged through the GSM.

Firstly, the encryption algorithm needs to be decided for encrypting the real time voice. In this work, an open source Advanced Encryption Standard was implemented in Java. After the choice of the cryptographic algorithm, the questions related to the key generation and sharing need to be addressed. It would also be necessary in future work to match the voice generated after decryption with the one that was encrypted.

Another important fact that needs to be considered is that no open source Android application exists that provides encryption of real time voice over the GSM. This is primarily because GSM uses a lossy algorithm, which makes the encryption of real time audio extremely difficult.

The lossy algorithm is based on the Linear Predictive Coding (LPC). The LPC helps to predict the upcoming voice by analyzing the previous voice samples. The LPC filter literally filters out all of the real time voice encryption. This is because as per its model, it considers all of the encrypted data as noise. So, any step forward in this respect should incorporate a method of cheating the LPC filter.

One method available includes the use of GSM codebooks [4]. The basic proposition, in this case, is to split the encrypted voice into three parameters before transmission over the GSM. The three parameters comprise the pitch energy and Linear Spectral Frequency (LSF).

Another question that needs to be addressed is the hardware implementation of this work. GSM communication has a well-established hardware, which gives us an indication towards the fact that this work may be hindered in future due to the hardware limitations. In such a case, it may become necessary to have customized hardware for secure voice communication.

7. Recommendations. Based on the results of this project, the first step forward is to create a single Android application combining all the different steps indicated in Fig. 1. However, there would still be need to proceed further for real time voice encryption.

A major focus of this project was on the Android Operating System (OS), which is why all of the work was carried out with the help Android Studio and NetBeans. Both are Java based platforms. The reason behind choosing the Android OS was mainly that it provides an open source platform, which not only includes the liberty of compiling applications as per one's desires and requirements, but also the kernel of the OS is available. A more invasive method of achieving the target can be the exploration and development of the Android OS in Linux/ Ubuntu. In the process, additional security features may be added with respect to the voice communication over the GSM network. This may give rise to a customized Android OS for particular GSM users.

Acknowledgment. This work needs to acknowledge the tireless efforts all of the worthy instructors of the College of Aeronautical Engineering, Pakistan Air Force Academy Asghar Khan, where this project received its due share of attention and prospered in a smooth fashion. Timely feedback and critique from the faculty of the College of Aeronautical Engineering gave an entire new dimension to this paper.

REFERENCES

[1] Imai12, H., Kobara21, K., & Morozov21, K. (2006). On the possibility of key agreement using variable directional antenna.

[2] Hellwig, K., Vary, P., Massaloux, D., Petit, J. P., Galand, C., & Rosso, M. (1989, November). Speech codec for the European mobile radio system. In Global Telecommunications Conference and Exhibition'Communications Technology for the 1990s and Beyond'(GLOBECOM), 1989. IEEE (pp. 1065-1069). IEEE.

[3] Khan, S., Pathan, A. S. K., & Alrajeh, N. A. (Eds.). (2012). Wireless sensor networks: Current status and future trends. CRC press.

[4] Ozkan, M. A., Ors, B., & Saldamli, G. (2011, December). Secure voice communication via GSM network. In Electrical and Electronics Engineering (ELECO), 2011 7th International Conference on (pp. II-288). IEEE.