Available online www.jsaer.com

***Journal of Scientific and Engineering Research*, 2019, 6(8):83-91**

Research Article

ISSN: 2394-2630
CODEN(USA): JSERBR

Enhanced Data Security in Mobile Text Messaging Using Firebase

***Karim Usman[1], Patrick Obilikwu[1], Anigbogu Gloria Nkiru[2], Adeyemo Mary[1]**

[1]Department of Mathematics and Computer Science, Benue State University, Makurdi, Nigeria
kusman@bsum.edu.ng
[2]Department of Computer Science, Nwafor Orizu College of Education, Nsugbe
anigbogugloria@yahoo.com

Abstract This study is focused on how to improve on the insecurities involved in sending text messages between users of mobile phones particularly Android phones. A text message is one of the major means of communication in mobile phones both within a long and short distance but since messages sent/received are always in plain and readable text, the messages are vulnerable and the content can be read by anybody that has access to the mobile phone. To achieve the goal of securing text messages, a mobile application was developed for communication between users through text messages. The mobile application was developed using Android Studio. The messages are secured using a cryptographic algorithm known as Advanced Encryption Standard (AES) which is a very secure, fast and reliable algorithm when implemented correctly. AES is used to encrypt and decrypt messages sent and received to avoid unauthorized access to the content of the message i.e. only the sender and the recipient will know the content of the message. The messages sent/received by users of the application is always in form of a cyphertext and can only be converted to a plaintext when the valid secret key used with the encryption algorithm (AES) to encrypt the message is entered. The secret key used for encrypting the message must be the same key used to decrypt the message. Also, Firebase is used to authenticate users of the application and serve as a real-time database where both the messages and secret keys are saved, retrieved and shared. Using the mobile application developed, users can send and receive encrypted message thereby preventing third parties from accessing the content of the message.

Keywords AES, Firebase, Android, Text message, authenticate

Introduction

In our daily lives, we communicate or share information either through physical conversions or electronic conversations. For most of human history, if you want to send a message, you have to deliver the message physically; shouting worked well for neighbours but as the distance increased, there was a need for new solutions. Some of the solutions were letters delivered by servants or messengers. Messenger pigeons were also bred and used for sending messages while some people rode on fast horsebacks to send and deliver messages. In the 1800s, the electronic telegraph system was established, after that, there were telephones, radios and televisions. Today we have mobile phones that we can use to make calls, send emails or share information/communicate on social media. This means that most of this information is being kept electronically now.

Apart from food, clothing and shelter being the basic needs of humans, security is also a basic need. There is a significant change in our society and daily lives brought about by the digital information revolution from analogue to digital conversion to the latest and sophisticated applications. The introduction of this development has made communication very easy but it has also made data and information to a larger extent vulnerable to unauthorized users, hackers or spammers to infiltrate the privacy of individuals, institutions and organizations.

This means the advantages provided by the digital information environment have brought about new challenges as well as new opportunities for innovation. Therefore, security and the fair use of data, as well as securely delivering or storing the data contents are very important yet challenging topics [1]. Mobile phones have become an integral part of the modern world, providing human connectivity in a way which was never possible before [2]. Researchers at the World Bank in 2012 estimated that nearly 3 in 4 people worldwide have access to a mobile phone. Over 95% of the global population is now covered by a mobile cellular signal. This means that the levels of information and communication technology access, use and skills continue to improve all around the world. In 2020, the number of smartphone users worldwide is projected to reach 2.87 billion, up from 2.1 billion in 2016. The ever-growing number of mobile phone users has provided a wide platform for both corporate organizations and government institutions to provide services to their clients. Mobile phones are very handy devices and are widely used by people around us for day to day functionalities. People are becoming more dependent on mobile phones for performing important functionalities like messaging, recording events, prepare and save daily and monthly budgets, bank transactions, etc. As people depend more on phones for faster processing, many sensitive data are stored on the phone and a considerable amount is also transmitted to the server while some are shared with other people. People have become very attached to their mobile phones; it is the first thing they check in the morning and the last at night.

Text messaging is one of the major features that mobile phone users have begun to fully exploit in recent years. Short Message Service (SMS) is a text messaging service component of most telephone, internet, and mobile-device systems. It uses standardized communication protocols to enable mobile devices to exchange short text messages. SMS was the most widely used application in 2010 with an estimated 3.5 billion active users or about 80% of all mobile subscribers. Mobile phone users can use SMS to send or receive personal messages, school activity alerts, stock alerts, email notifications, social media notification, job dispatches, and so on from either a single person or several persons. The short messages are sent in plaintext and binary format and can be easily intercepted and altered with any of the existing cracking tools and this loophole provides a great opportunity for hackers to exploit. An example is Spy applications such as MobiStealth, Spyzie, Highster Mobile, XNSPY, FlexiSPY, mSpy and SpyEra that can intercept text messages and record conversations without the knowledge of the user. Therefore, the security issue of SMS is still an open challenging task. Nevertheless, the only general approach to sending and storing data is to use some form of encryption and different methods such as steganography, cryptography, etc. have been used. This work aims to develop a text messaging application that is secure using Advanced Encryption Standard (AES) cryptographic algorithm which will allow users to encrypt and decrypt the text messages easily and efficiently and provide a secure means of sharing the secret key. The objectives of the study are: to develop a prototype that encrypts text messages before they are sent, to employ an AES encryption technique in the process of the transmission of the message and to provide a secure way of sharing the secret key for decrypting the message using Firebase platform.

Related Works

Sharad, K. V. and Deo, B. O. [3] focused their work on enhancing mobile SMS security using Caesar cypher and one-time pad. This is a substitution cypher in which letter in the plaintext is replaced by a letter some fixed positions down the alphabet and after that they used the one-time pad technique to perform the encryption again. One of the drawbacks of their method is that it requires a pad that is the same length as the message to be encrypted (i.e. the key is as long as the plaintext). The key also has to be genuinely random for this method to be effective and this is hard to achieve for large keys. Also, they did not consider the security of the key.

Croft, N. J. and Olivier, M. S. [4] used an approximate one-time pad to alleviate SMS security vulnerability. The practical difficulty of using this method is that the key size must be equal in length or longer than the message being encrypted and the key bytes can not be reused. This means that even for a two-way exchange of messages, each party must have a sufficient supply of key material on hand so they do not run out. Also, a secure channel will be needed to exchange the OTP but this was not considered in their work.

Rahman, A. et al. [5] developed an Android and web application for securing messaging system using a monoalphabetic substitution algorithm. For monoalphabetic cypher to be used, the key has to be memorized. One of the major weaknesses of this algorithm is that although the letters themselves change, their frequency

does not. Therefore, any enthusiastic cryptographer can crack the code using frequency analysis table of the original plaintext but this can not be done if the AES algorithm is used for the encryption.

Hazem M. et al. [6] focused on SMS/Multimedia message using an encryption (Blowfish) system. The proposed technique encrypts SMS with 16-round Feistel cypher and uses large key-dependent S-boxes. However, Blowfish has some weaknesses in the decryption process over other algorithms in terms of time consumption and serially in throughput. Also, Blowfish is vulnerable to birthday attack.

Huseyin, B. and Resul, K. [7] developed SMS encryption using the RSA encryption algorithm. Their application was tested with different key sizes. They did remarkable work in reducing the complexity of RSA encryption algorithm, but a major drawback of their application is that RSA can be very slow in cases where large data need to be encrypted when compared to AES.

Tarek, M.M. et al. [8] developed a hybrid compression encryption technique to secure SMS data. Their technique compresses the SMS to reduce the size and encrypts it using the RSA algorithm. Although they tried to reduce the SMS size, the RSA algorithm is slower and less secure than AES.

Sri, R. et al. [9] used Elliptic Curve Cryptography (ECC) where domain parameters are the constants. Though it uses less bit size and low processing power than RSA, the method is much slower than AES. Also, the ECC algorithm is more complex and difficult to implement and this increases the likelihood of implementation errors, thereby reducing the security of the algorithm.

Gurpreet and Supriya [10] presented a paper with a detailed study of the popular encryption algorithms such as RSA, DES, 3DES and AES. From the research done, there was a conclusion that the AES algorithm is more efficient in terms of speed, time, throughput and avalanche effect.

Muhammad, N.R. and Adeel, I. [11] carried out one of the most remarkable research in their study of AES algorithm in Android SMS but one of the most important things they ignored in their study is the secure means of sharing the secret key. This work, therefore, serves as a drive to investigate how to improve the means of securely sharing the secret key.

The Proposed Model

In this work, an Android application is developed to improve on the issue of a security problem when using text message for communication. In this system, the user installs the application on his/her mobile phone which has to be an Android device (the user must be connected to the internet). The user is authenticated and verified using his/her phone number. After the authentication, the user creates a profile and a password which will be used to access the secret key. Once the profile is created, the user can send messages to any other registered user using their username. The message is sent by clicking on the message button, the recipient's username is entered in the field provided for the recipient and the message is typed in the field provided for entering the message. Once the send button is clicked on, the application generates the secret key and uses the secret key and AES algorithm to encrypt the message; this means that the message is transmitted in an encrypted form.The message is displayed as a jumbled text both for the sender and the recipient. The key generated together with the message is stored in the Firebase database. When the recipient receives the encrypted message, he/she will enter the password to retrieve the key before the message can be decrypted. If the password is valid, the key is retrieved from firebase and the cyphertext is converted to a plaintext which is displayed to the user. If the password is invalid, the text remains encrypted and an error message is displayed indicating that the password is invalid. Also, the user can delete messages from the application.

Methodology

The methodologies employed in this research to meet the objectives are: UML diagrams (use case diagram) used to make the coding process of the application easier, Android Studio used as a platform to write the code for the application, Android device used to download and test the application and Firebase server and Firebase database platform to store and share the secret key and messages between users of the application. Firebase is a platform used for several development purposes which also includes the hosting, monitoring and maintenance of mobile and web applications. Firebase provides UI libraries and Software Development Kit (SDK) to implement authentication of users across platforms using email id, username and password or other federated identity

provider integration such as Google and Facebook. It allows the implementation of various sign-in techniques to allow users to log in to applications. Some of these sign-in techniques are a phone, Google, Facebook, Twitter, Yahoo, GitHub, Microsoft, Email/Password, etc. In this work, the 'phone' sign-in technique is used to authenticate users of the application. Firebase has a real-time database and NoSQL cloud-hosted database where data can be stored as JSON. This is the foremost advantage of the firebase. If this feature is used, there will be no need to create your database or API. Firebase handles all the components that usually come along with creating a backend for applications. It gives an adaptable, expression-based rules language to define how your data should be organized. When using Firebase real-time database, every time a data is updated, the database stores the data in a cloud and simultaneously notifies all the other devices in milliseconds. This simply means that it synchronizes data between all the users in real-time and this makes it easy for users to collaborate. In other, for the database to function well in real-time, the database handles offline storage. The two ways of handling offline storage are:

i. **Intermittent Offline Storage***:* This is used when the internet connection drops shortly. The real-time database SDK puts a local in-memory cache on the device so that the changes made when the internet connection dropped can be served through the cache when the connection is restored.
ii. **Long Term Offline Storage***:* This is used when the user turns off the internet connection. In this case, a persistent cache is enabled on the device. The persistent cache helps to store all the updates made by the user while offline and the data is updated by the database once the user is online. The updated data is merged with the data in the database and the merged update is synchronized to all the devices.

Also, Firebase has a set of security and rules authentication which specifies who has access to particular data. These rules are securely stored with the real-time database on Firebase server and hosted on the cloud. By default, Firebase authentication is required in the database rules and full read/write permission is only granted to authenticated users. The default rules ensure that your database is accessible by only authenticated users. SDKs for Android, iOS and JavaScript are available on Firebase to ease the implementation of both the authentication and the real-time database. The application developed in this work is connected to Firebase using Firebase SDK.

Results and Analysis

For our study, the researchers used a laptop with the following processor configuration: 2GB of available disk space, CPU core i5, 1280 x 800 screen resolution, Windows 10 and 64-bit Operating System. An Android device with an internet connection was also used to test the application. The application was designed to make communication through text messages on mobile phones easy, fast and reliable. It provides Graphical User Interface (GUI) which enables users to interact with the application easily. It is also designed to be flexible to make it easy for the system to be modified. Some of the screenshots of the application are shown below.

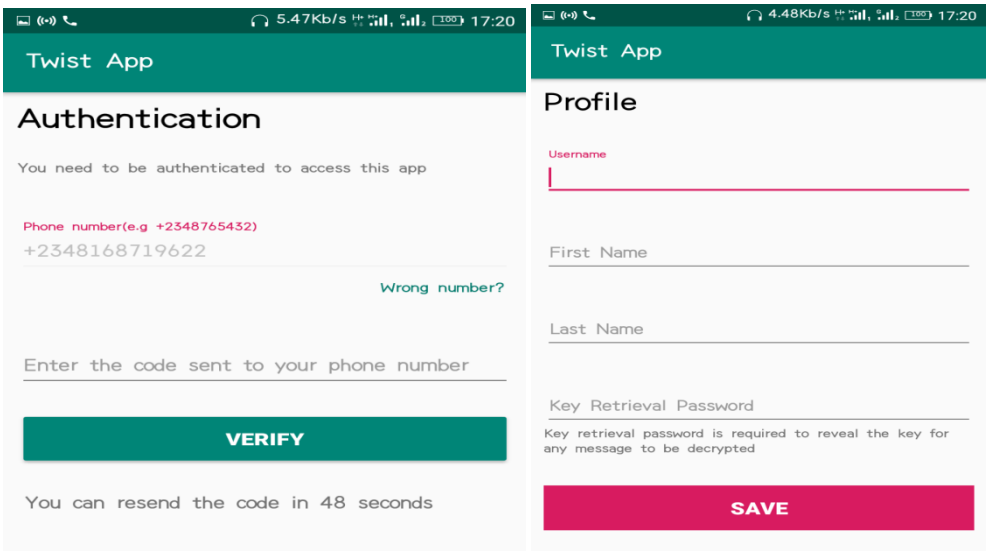

Figure 1: User authentication and profile creation

Once the application is installed, the authentication page loads. The user must have an internet connection on the device used. Also, the user is required to enter a valid phone number before they can use the app. After the user enters the phone number, a verification code is sent as a text to the user. The code is entered in the space provided in the figure above. The next page is the profile creation page. The user is required to enter the preferred username, first name and last name. The username created is what will be displayed when the user sends a message and it is also entered as the recipient instead of the phone number when sending messages. Also, the user has to create a password that will be used to retrieve the key for decrypting messages from the firebase database where the key is stored after being generated by the application. This password is kept secret and known only to the user of the app. Once the profile is created, the user can now send and receive messages.

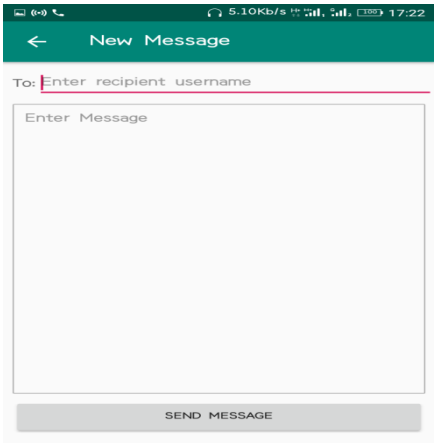

Figure 2: Compose Message

The user types the username of the recipient and the text message and clicks on send. After clicking on send, a prompt is displayed requesting confirmation from the user. If the user clicks on 'No', the message will not be sent and if the user clicks on 'Yes', the message will be sent and information will be displayed on the user's screen informing the user that the message has been sent successfully.

In the backend, once the user confirms that the message should be sent, the secret key is generated by the application and each key generated is unique and can only be used with that particular message. This means that every message has its key and the key of one message cannot be used to decrypt another message. The application is designed to generate the secret key in other to secure the message more since the user will not have to call the recipient to disclose the secret key to him/her. After the secret key is generated, it is used to encrypt the message using AES algorithm.

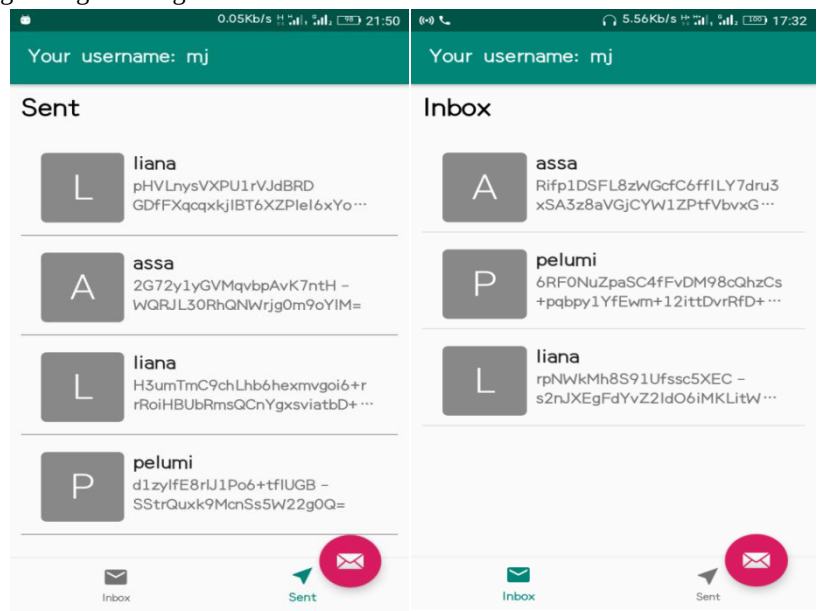

Figure 3: User's sent box and inbox

These pages display all the messages sent and received by the user of the application. All sent and received messages are displayed as encrypted messages and can only be viewed as plaintext if it is decrypted.

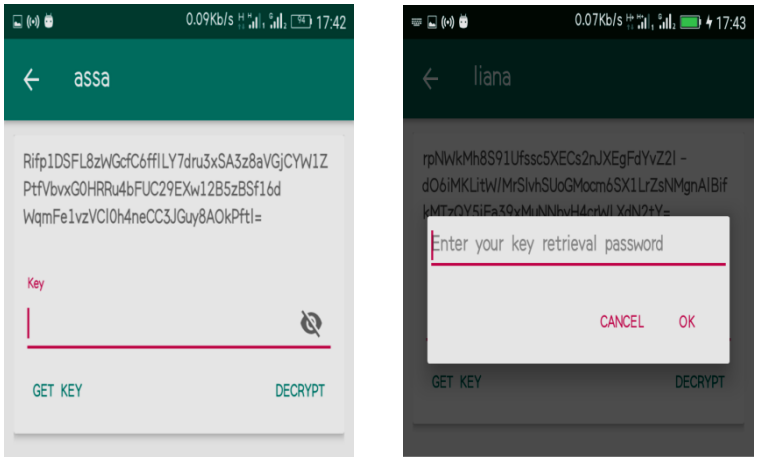

Figure 4: Read message

To read the content of a message, the user has to click on the message. The message will still be displayed as encrypted text. Since the message is displayed in encrypted form, it has to be converted to plaintext. To convert to plaintext, the user clicks on 'Get Key' and enters the key retrieval password created while updating the profile. If the password is valid, the key is retrieved from firebase and used to decrypt the message, if invalid; an error message displays 'invalid key password'.

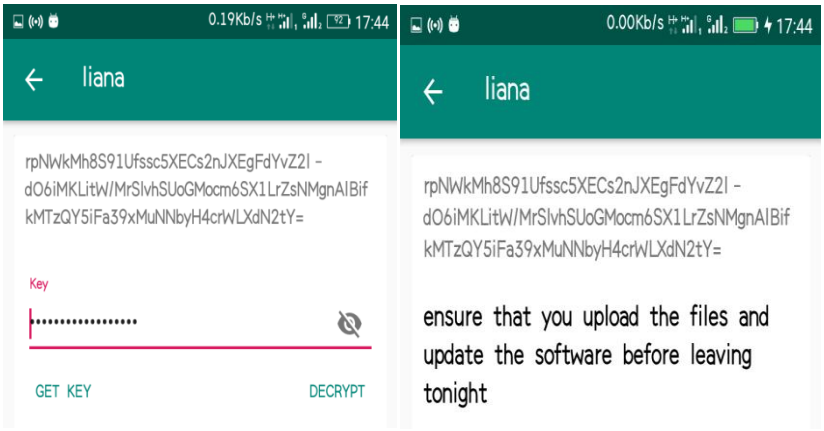

Figure 5: Message converted to plaintext

The message can only be decrypted if the key retrieval password is valid. Once the key has been retrieved, it displays in the space provided for key. The user can now click on 'Decrypt' to view the plaintext. The plaintext is displayed after the message is being decrypted and the user can now see the text in a readable format.

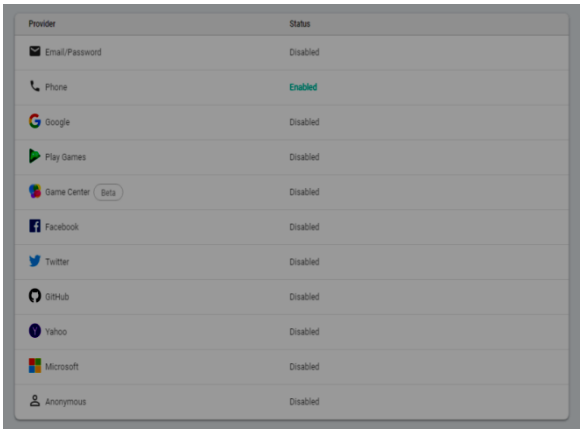

Figure 8: Firebase authentication

This Figure shows the different methods of sign-in that can be used for the application. As seen in the Figure above, the ‘phone' is enabled; hence it is the only method that can be used.

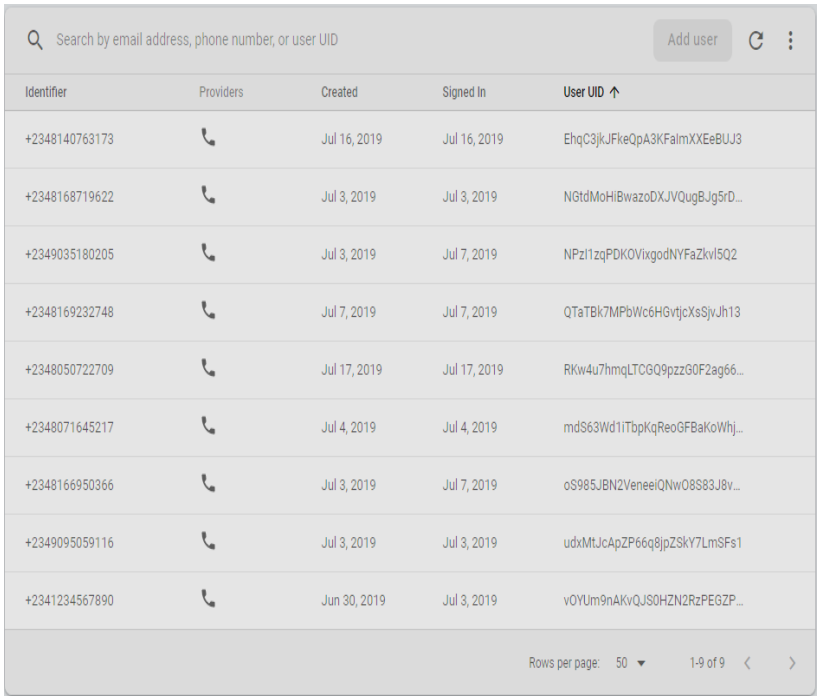

Figure 7: Firebase users tab

This Figure shows the list of users authenticated to use the application stored in the firebase database.

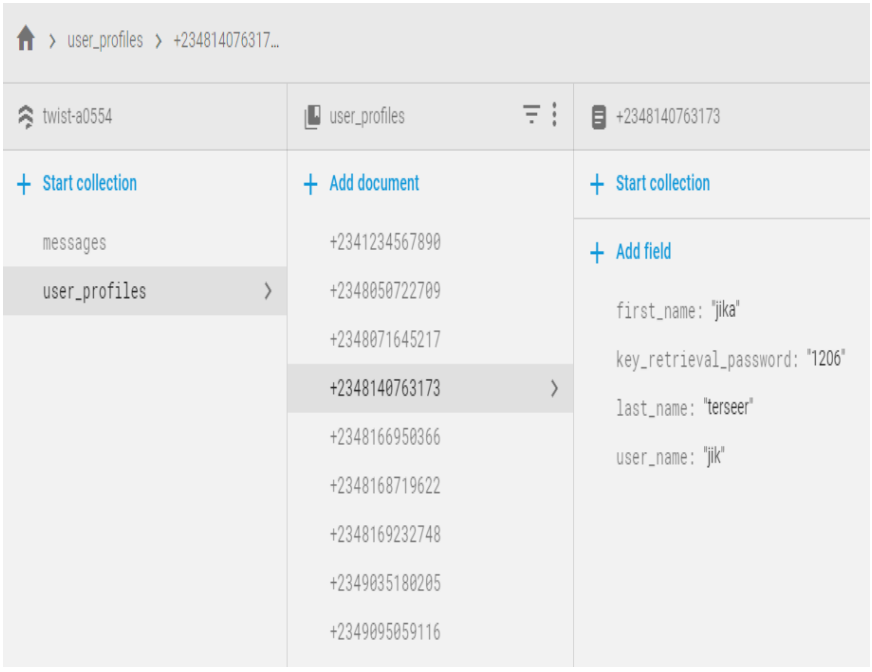

Figure 9: User profile database

This displays the list of the user profiles created on the application. It displays the first name, last name, username and key retrieval password associated with each user. Each user is uniquely identified with their phone numbers.

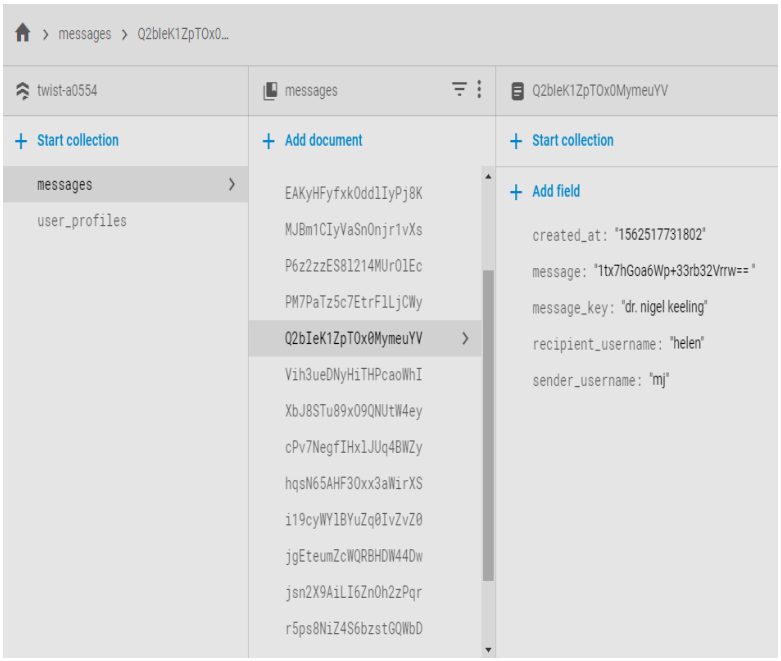

Figure 10: Messages database

The Figure above shows the list of messages sent and received by users saved in the firebase database. It shows details including the message sent/received, sender's username, recipient's username and the message key and as seen in the figure, the messages are all encrypted.

Conclusion

Communication is one of the most important things in life. Imagine a world where we live without any means of communication? I am very sure it is unimaginable as people would have no way of expressing their thoughts, ideas or feelings to others. Communication is indispensable and this is why even animals communicate among themselves using body languages, touch, smell, noise, etc. Knowing how crucial communication is, phones were developed to ease the transfer of information between people. This is why the basic and general characteristics of every phone are making calls and sending text messages.

Because sending text messages is one of the major means of communication, the development of an application to secure the information transferred is of great importance. This is why the importance of this work cannot be overemphasized as it meets this need for secure communication using text messages. Although this application does not provide 100% security (which is impossible as humans are the greatest threat to security), it helps to mitigate the issue of insecurity in text messages to a very large extent. This project has achieved its objective that was set to develop an improved mechanism of the text messaging application. The application developed was able to apply the advanced encryption concepts to encrypt and decrypt the text messages. The experimental results showed that the system can encrypt and decrypt messages. Unlike the normal text message that is sent in plaintext via GSM network, messages sent via this app remain encrypted during transmission.

Therefore this work has helped in developing an application that can be used not only by individuals but also within industries, public and private enterprises, organizations, institutions, etc. to secure vital information that needs to be transmitted through text messages within a long and short distance.

AES algorithm is considered to be the most secured cryptographic algorithm now but it becomes useless if the secret key can be gotten by anybody without authorization. This work considered the security of the secret key and this is why the application was developed to generate the key instead of the user so that the user will not need to call the recipient to share the secret key with them. Although this has helped to improve the security of the text message, further studies should also focus on not only the algorithm to be used for encryption and decryption but also how the secret key can be shared between the sender and recipient more securely. Also, further works can be focused on improving compatibility with other mobile devices such as devices running on

Windows and iOS. Finally, the application can be modified to send multimedia files such as videos, audios and images.

References

[1]. Blaz, M. and Igbor, B. (2012). Mobile Devices and Corporate Data Security: International Journal of Education and Information Technologies, 6(1).

[2]. Jeff, B., Bill, S. and Ron, V. (2007). SMS: The Short Message Service. Retrieved on April 23, 2019, from https://www.researchgate.net/publication/2962066_SMS_The_short_message_service

[3]. Sharad, K. V. & Deo, B. O. (2014). An Approach to Enhance the Mobile SMS Security. Journal of Global Research in Computer Science 5(5), 1-6.

[4]. Croft, N. J. & Olivier, M.S. (2005). Using an Approximated One-Time Pad to Secure Short Messaging System (SMS). Southern African Telecommunication Networks and Applications Conference 20051. 71-76.

[5]. Rahman, A. et al. (2016). Development of Cryptography-Based Secure Messaging System. Retrieved on July 9, 2019 from https://www.researchgate.net/publication/312047086 _Development_of_Cryptography-Based_Secure_Messaging_System

[6]. Hazem M. et al. (2013). A New Mobile Application for Encrypting SMS. Abstract Retrieved on April 23, 2019 from https://www.researchgate.net/publication/307545492_A_New_Mobile_Application _for_Encrypting_SMS_Multimedia_Messages_on_Android

[7]. Huseyin, B. & Resul, K. (2015). Secure SMS Encryption Using RSA Encryption Algorithm on Android Message Application. Abstract Retrieved on April 23, 2019 from https://www.researchgate.net/publication/298298027_Secure_SMS_Encryption_Using_RSA_Encryption_Algorithm_on_Android_Message_Application

[8]. Tarek, M.M. et al. (2010). Hybrid Compression Encryption Technique for Securing SMS. International Journal of Computer Science and Security, 3(6), 473-481.

[9]. Sri R. et al. (2013). Securing SMS using Cryptography. International Journal of Computer Science and Information Technologies, 4(2), 285-288

[10]. Gurpreet, S. & Supriya (2013). A Study of Encryption Algorithms (RSA, AES, DES and 3DES) for information security. International Journal of Computer Applications 67(19).

[11]. Muhammad, N. R. and Adeel, I. (2018). Development of a Secured SMS Application Using AES on Android Platform. Retrieved on June 12, 2019 from https://www.researchgate.net/publication /323141357_Development_of_a_Secure_SMS_Application_using_Advanced_Encryption_Standard_AES_on_Android_Platform