**PROJECT SUMMARY**

**Introduction**

Short Message Service (SMS) also known has text message is a communication service standardized in the Global System for Mobile (GSM) system. It supports up to 160 7-bit characters or 140 8-bit characters and the service works on a store-and-forward basis. SMS messages are not sent directly to the recipient; instead they travel through several nodes before reaching the recipient. This leads to serious security and privacy problems. There are two types of encryption and decryption mechanisms, symmetric (privet) key cryptography and asymmetric (public) key cryptography. Encryption and decryption provide powerful tools for protecting sensitive communications over a public network, but it adds an overhead in terms of additional computation. This additional computation requirement reduces the applicability of cryptography on embedded devices (mobile phone) because mobile phone has limited resources such as computational power, battery life and memory.

Currently, mobile SMS does not provide any specific security feature. Because of this, one cannot be sure to send and receive confidential and sensitive data such as bank account information and patient information safely. By implementing secure SMS system, many SMS security threats such as Man-in-the-middle attack, SMS-spoofing, interception and other related attacks are greatly reduced. The proposed secure end-to-end encryption will ensure the message to be read only by the right person. The privacy and confidentiality of SMS message content can be protected. As clearly known, mobile phone has limited resources (processer, and memory). Because of this, identifying the technique that needs minimum computational resource with acceptable quality is a major issue in mobile application.

The aim of this research is to identify the appropriate encryption method for mobile text messaging application and develop SMS security mechanism on android enabled mobile. The objectives are to:

1. Study and identify the appropriate encryption/decryption algorithm for mobile application;
2. Study and identify the appropriate security technique for end-to-end text message security;
3. Design the architecture of secure text message using android studio;
4. Develop a prototype for secure text messaging application which enables the user to send and receive confidential information using any android enabled mobile phone; and
5. Evaluate the performance and quality of service after implementation of the secure system.

**Related Works**

Many of SMS security solutions are based on symmetric key cryptography (Rohan, Sanket, and Priyanka, 2012) due to limited infrastructure of mobile devices: such as limited memory, processing capacity and power supply (Raghavendra, Sunanda, and Maruthi P, 2011). Most of the proposed symmetric encryption involves an assumption that the communicating parties has prearranged shared secret key in order to encrypt the message.

Another paper by Hassan et al. (2011) proposes AES encryption technique to exchange secure SMS. This paper describes a system for securing SMS messages during and after its transmission over mobile network. As mentioned in this paper, the system can send encrypted messages and allow users to encrypt/decrypt messages for personal usage. The system depends on secret key embedding, where the message’s secret key is distributed inside the cipher text after message encryption process. Secret key embedding is used for checking the correctness of a decryption key which is entered by the user. As mentioned in the paper, this scheme saves time and space as there is no need for a database to store the secret key related to each message. But this system has numbers of limitations; the first one is the receiver must remember the secret key of associated sender because, the secrete key is not stored on the mobile phone. The other limitation is, it increase the size of the message because of the embedded secret key on the encrypted message. As mentioned in the paper embedding and extracting secret key is not implemented and not tasted.

The paper by Singh et al. (2012) describes an approach to android or mobile application that allows the user to share textual information via SMS without being intercepted by any unauthorized user. A cryptography technique had proposed which avoids the leakage of clear messages being transferred from one end to the other, by encrypting it by using an already exchanged secret key. As described in the paper, the secret key exchange is done using the Diffie-Hellman key exchange algorithm, and AES-128 is used as the encryption/decryption algorithm. The system supports only android mobile phones.

Anuar et al. (2008) proposed a framework for SMS by using both symmetric (AES) and asymmetric (RSA) cryptography. In this system the message classified as normal SMS (no encryption), internal SMS (symmetric encryption), and confidential SMS (asymmetric encryption). The paper proposed m–PKI to provide public key encryption to the mobile SMS. This approach allows the end-user to send private and classified message via SMS. In this solution the content of the SMS can be encrypted by using AES or by using RSA based on the classification.

Hybrid cryptographic scheme had been proposed by Al-bakri et al. (2010) to solve security limitation of SMS messaging. As mentioned in the paper, they had used the combination of asymmetric (NTRU) and symmetric (AES-Rijndael) cryptography algorithms to achieve more robust functionality. One of the limitations of this system is, it requires two pair of keys. In other word the keys in NTRU are not interchangeable. This means the user needs NTRU Encrypt key pair for encryption or decryption processes and another NTRU Sign key pair for digital signature processes. The other drawback of this application is the cipher text size produced after encryption is large. This system produces 1757 byte from 140 byte SMS. Thus, using this application is not advisable for SMS based applications.

In general symmetric cryptography solution for SMS security had some limitations; first it needs secure channel for key exchange or the security of the message is strongly depends on the secrecy of the key. One other limitation is the sender and the receiver must maintain large numbers of key approximately n(n−1)/2 keys where n is numbers of users. Also, this approach not supports some security requirements of SMS such as non-repudiation.

Some drawbacks of asymmetric cryptography based approach are: it needs more computational resource such as, memory and processer power than symmetric cryptography, also in asymmetric cryptography size of cipher text greater than the original message. Therefore, it is not advisable encrypt the whole SMS content using asymmetric key encryption.

In this final year project, an effective encryption based solution is proposed. In the proposed solution, symmetric (Blowfish) cryptography is to be used to encrypt/decrypt the SMS content and asymmetric (ECC) is to be used to transfer the secrete key which is used to secure the SMS content. To solve key transmission problem of symmetric key cryptography, in the proposed solution, the secret key is to be transmitted together with the encrypted SMS content in a secure manner. This solves the requirement of large key size. To do this, the proposed system generate random secrete key, encrypt it by using ECC and transmit to the receiver together with the SMS content at a time, thus, in this system, a single user needs only one private-public key pair. As observed in the study of related works, encrypting the whole content of SMS message using asymmetric cryptography is not advisable. To reduce this problem, the proposed system only encrypt the secrete key by using asymmetric (ECC) cryptography.

**Design Methodology**

The SMS mobile app contains the following sub systems. The Encryption/Decryption Subsystem**;** which contains two type of encryption, the first one is symmetric (Blowfish) cryptography and the second is asymmetric (ECC) cryptography. Blowfish has a better performance than other common encryption algorithms in different aspects such as speed, memory usage and throughput. Key Encryption/Decryption Subsystem**;** the main task of Key Encryption/Decryption subsystem is to encrypt one time password (OTP) which is used as secret key for symmetric cryptography. This encrypted onetime password is generated by using ECC algorithm. ECC with 160-bits key size is applied. ECC with 160-key size have equivalent security level with RSA 1024-bit key size.

Combiner/Extractor Subsystem is the one that concatenates the encrypted onetime password and encrypted SMS message at the sender side in order to produce secure SMS payload. Also at the receiver side it used for extraction the two parts of the secure SMS payload. Combiner: the proposed Secure SMS payload is the combination of encrypted one time password and encrypted SMS message. The main function of this component is concatenating the output of Key Encryption subsystem (encrypted onetime password) and the output of Message Encryption subsystem (encrypted SMS message). The extractor works in such a manner that the onetime password is transmitted together with the SMS message to the receiver. Before decrypting the received secure SMS, the recipient need the secrete key which is used to decrypt the secure SMS payload. As mentioned, this onetime password sent together with SMS message. Then, at the receiver side, this component extracts the onetime password from secure SMS payload and this extracted onetime password is decrypted by using receiver’s private key.

It was decided to implement the android messaging system using Java programming language on android studio. Among the reasons are:

* Familiarity;
* Object Orientation;
* Level of abstraction: Java provides a good level of abstraction and it includes rich set of APIs.
* APIs: Java provides SMS messaging APIs in the WMA (Wireless Messaging APIs) set. It also provides a set of APIs to build useful user interfaces.
* Availability: Most mobile phones being sold on the market includes built in Java virtual machine.
* Portability: Java ME makes the application portable on to any java enabled phone. For these reasons we have chosen java as the programming language for the implementations.

In order to develop the secure SMS messaging application Java Wireless Toolkit development kit was integrated with android studio, android studio was used as development environment. The message sender and receiver application uses the NetBeans Mobility Pack 3.0 to simulate mobile phone environment. Since the mobile application is developed using Java, it should be able to run on any mobile phone that has Java Virtual Machine (JVM).

To enable the application to send and receive SMS message on the mobile phone, the J2ME Wireless Message API (WMA) library was used.

**Results and Discussion**

The development of this work was carried out on a HP GS78 Laptop Computer with the following specifications;

* 1 Terabytes Hard disk
* 8 Gigabytes RAM (Random Access Memory)
* 4 Gigahertz Processor Speed (Intel Pentium Dual Core).
* 64-bit Operating System (Windows 7)

The deployment and testing was carried out on HTC M9 mobile phone.

On first installation the user is asked to create pin so as to secure the app. After a pin has been selected the main layout (home page) is shown. The home page of the secure SMS app shows an inbox tab and a sent tab and a send new message button. The Inbox Tab shows a list of inbox (received) messages. The Sent Tab shows a list of sent messages. The app takes over the SMS functionality of the mobile phone; this means all text messages sent to the mobile phone through the phone number enters the secure SMS app. Only messages sent to the phone from another phone that has this secure SMS app would be encrypted, other text messages sent from the default SMS app will enter the app, but would not be encrypted. All text messages sent from the secure SMS app would be delivered to the recipient phone as encrypted text messages, but only the secure SMS app on the recipient phone can open the encrypted messages, default SMS app would not be able to open the encrypted messages. When the Create New Message is clicked, a screen is brought forward and a prompt for the use to input the destination phone number and a message content.

There is also a button at the bottom of the screen to encrypt the message. When the button is clicked a key is generated using ECC encryption and the key is used to encrypt the message using blowfish encryption and then encoded in base64 so as to pass over the network and the encrypted message is sent over the network. After clicking on the message the user wants to view, a screen is brought forward and the encrypted message is shown. At the bottom , there is a button named “Decrypt”, this decrypts the Message content for view,  if the message is of an encrypted type, the plain text is shown but if not,  an error message is shown.

For an encrypted message, the key is generated at the point of creation of the text message. At the recipient’s end, the key is extracted from the content and used to decrypt the main body of the message and it is displayed to the user. When the “Create New Message” is clicked, a screen is brought forward and a prompt for the use to input the destination phone number and a message content. There is also a button at the bottom of the screen to encrypt the message. When the button is clicked a key is generated using ECC encryption and the key is used to encrypt the message using blowfish encryption algorithm and then encoded in base 64 so as to pass over the network and the encrypted message is sent over the mobile network. On every run of the app, the user has to input the security pin created so as to be able to access the app.

**Conclusion and Recommendations**

The study is motivated by the fact of ensuring privacy and security in SMS systems, but solution cost was also taken into account vigorously. Several relevant works were analyzed for gathering knowledge to accomplish this research. This is an application layer protocol and low cost approach in context of computational and implementation scenario. For any SMS service, this approach can be used to ensure privacy and security. However, the operators and handset /mobile phones providers need to be aligned with this approach and cumulative understanding can ensure secured messaging with high privacy.

In the proposed solution, Hybrid encryption was used, which is the combination of symmetric and asymmetric key encryption. Asymmetric key encryption (ECC) used to encrypt the one time password which is used as secret key to symmetric cryptography. Symmetric cryptography (Blowfish) was implemented for encrypt and decrypt the original SMS.

The designed system provides a secure communication platform between two mobile phones over GSM network. This system was implemented using android studio which runs with java programming language and therefore the mobile application software can be installed on any android operating system based mobile phone. Based on the results of this project, the following can be concluded:

1. Blowfish algorithm can be applied to applications based on Android SMS Cryptography.

2. SMS Cryptography Applications can be run on Android versions as low as version 4.2.2 (Kitkat), the Android version 4.1.2 (Jelly Bean), Android Version 3.2 (Icecream Sandwich), the Android version 3.0 (Honey Comb), Android Version 2.3 (Ginger Bread). Besides emulators, the application was also tested on real mobile phone with newer android OS versions 6, 7 and 8 such as Samsung, HTC, Tecno etc.

The proposed system can be easily adapted to support secure SMS messaging solutions for many applications such as SMS based (USSD) banking, SMS based health and SMS based voting. This solution is highly recommend to any commercial company and government organizations, which need confidential information transmitted over the GSM network. The work has been done by sending the shared or secret key, which is used to encrypt and to decrypt the SMS message, together with the message. To send the secrete key along with the message encrypted by using ECC asymmetric encryption mechanism

As we know one of the drawback of this asymmetric encryption is it produce larger size cipher or encrypted text than the original text. Thus, one of the future work suggested is to research on how the size of SMS content to be sent can be decreased as a single SMS message by using some compression technique on key encryption/ decryption part of the proposed system.

For further algorithm implementation, the study is also looking forward for possibilities that the ECC and Blowfish algorithms implemented can also be applied and implemented in different environments not only in SMS communication lines but also on email, documents, social media and other transactional data for privacy and security purposes. For further enhancement, the security framework may also be compared to other encryption algorithms